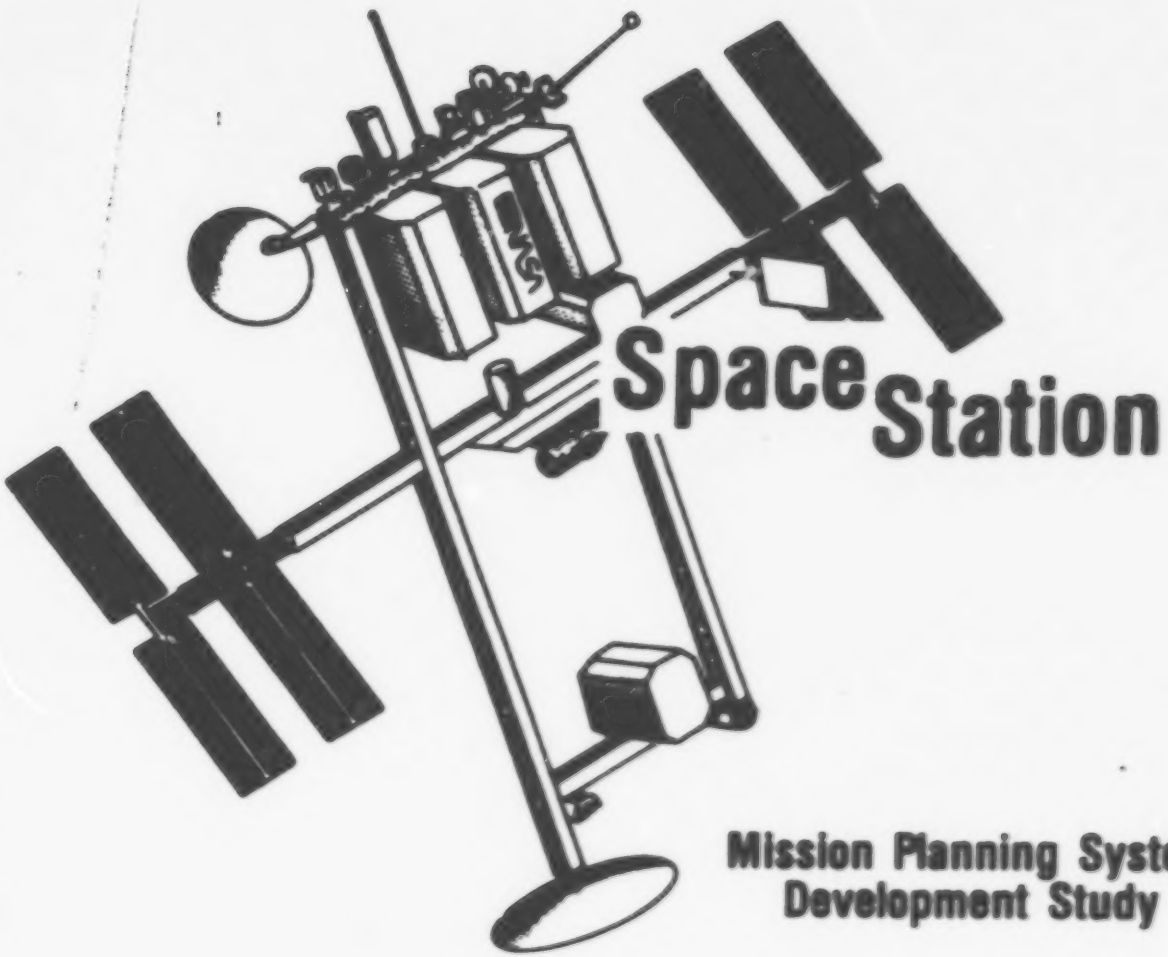
MARCH 1987

MDC W5188



Space Station

Mission Planning System Development Study

Final Report
Volume III - Software Development Plan

(NASA-CR-179203) SPACE STATION MISSION PLANNING STUDY (MPS) DEVELOPMENT STUDY. VOLUME 3: SOFTWARE DEVELOPMENT PLAN Final Report (McDonnell-Douglas Astronautics Co.) 74 p Avail: NTIS HC A04/MF A01 CSCL 22A G3/12

N88-10049

Unclas 0069651

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY
HUNTSVILLE DIVISION

MCDONNELL DOUGLAS CORPORATION

MARCH 1987 MDC W5108

Space Station

Mission Planning System (MPS) Development Study

Final Report
Volume III - Software Development Plan

PREPARED BY: _____
W. J. Kiss
Project Manager

APPROVED BY: _____
E. M. Channing
Manager, Space Station Projects

PREPARED FOR THE NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, GEORGE C. MARSHALL SPACE FLIGHT CENTER, UNDER CONTRACT NO. NAS8-37275, EFFECTIVE DATE: 23 MAY 1986

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY
HUNTSVILLE DIVISION
P.O. BOX 1181
HUNTSVILLE, AL 35807

CONTENTS

			Page
	LIST OF FIGURES		iii
	LIST OF TABLES		iii
Section 1	INTRODUCTION		1-1
	1.1 Purpose		1-1
	1.2 Scope		1-1
Section 2	PROJECT TECHNICAL DESCRIPTION		2-1
Section 3	PROJECT ACTIVITIES		3-1
	3.1	Overview	3-1
	3.1.1	Major Activities and Formal Reviews	3-1
	3.1.2	Documentation	3-1
	3.1.3	Definition of Computer Program Structure	3-4
	3.2	SS MPS System Requirements Definition/Design	3-4
	3.3	Software Requirements Definition	3-4
	3.4	Preliminary Program Design	3-5
	3.5	Detailed Design and Analysis	3-7
	3.6	Coding and Unit Testing	3-8
	3.7	Module Integration And Testing	3-9
	3.8	Computer Program Testing	3-10
	3.9	Unique Activities in SS MPS Development (Computer Program Level)	3-11
	3.9.1	Prototyping	3-11
	3.10	SS MPS Simulations and Performance Review	3-12
Section 4	DEVELOPMENT SCHEDULES AND MANPOWER REQUIREMENTS		4-1
	4.1	Methodology and Assumptions	4-1
	4.1.1	Lines of Code	4-1
	4.1.2	Software Cost Drivers	4-1
	4.2	Manpower Requirements	4-1
	4.2.1	SW Set Manpower Requirements	4-1
	4.2.2	Manpower Requirements Summary	4-4
	4.3	Schedules	4-11
	4.3.1	SW Set Schedules	4-11
	4.3.2	Top Level SS MPS Schedule	4-11
Section 5	SOFTWARE DEVELOPMENT PROCEDURES		5-1
	5.1	SW Techniques and Methodologies	5-1
	5.1.1	Structured Analysis for Software Requirements	5-1
	5.1.2	Structured Design and Software Top Level Design	5-1

5.1.3	PDL For Software Detailed Design	5-1
5.1.4	Standard for Code Development	5-1
5.1.5	Standard for Unit Testing	5-2
5.1.6	Standard for Module and Computer Program Integration and Testing	5-2
5.1.7	Walkthroughs	5-2
5.2	Software Development Library	5-2
5.3	Software Development Files	5-3
5.3.1	Benefits of the Use of SDFs	5-3
5.3.2	Responsibility for Maintaining SDFs	5-3
5.3.3	Creation and Maintenance of SDFs	5-4
5.3.4	Contents and Format of the SDFs	5-4
5.4	Documentation Formats for Informal Test	5-5
5.4.1	Unit Test Plan/Description	5-6
5.4.2	Unit Test Procedures	5-6
5.4.3	Unit Test Report	5-7
5.4.4	Module Integration Test Cases	5-7
5.4.5	Module Integration/Test Procedures	5-7
5.4.6	Module Integration Test Results	5-7
5.5	Design and Coding Standards	5-7
5.6	SW Development Tools	5-9

Appendix A - SW Functional Requirements A-1

Appendix B - COCOMO SW Set Cost Estimates B-1

FIGURES

Number		Page
2-1	SS MPS SW Hierarchy	2-2
4.1.2-1	Software Cost Driver Ratings (COCOMO Input Parameters)	4-5
4.2.1-1	Manpower Per SW Set Phase	4-6
4.2.1-2	Sensitivity of Estimated Manmonths to Various Cost Driver (Rescheduler)	4-9
4.3.1-1	SW Set Development Schedules	4-12
4.3.2-1	SS MPS Top Level Development Schedule	4-15
4.3.2-2	SW Set Phasing Criteria	4-16

TABLES

3.1-1	Required Formal Reviews	3-2
3.1-2	Required Formal Documentation	3-3
4-1	SW Set Groupings and Lines of Code Estimates	4-2
4.2.2-1	Manpower Requirements Summary	4-10
4.3.2-1	Manpower Loading Estimates	4-17

Section 1

INTRODUCTION

1.1 PURPOSE

The purpose of this volume is to present a software development plan for the definition, design and implementation of the Space Station (SS) Payload Mission Planning System (MPS).

This plan is an evolving document and must be updated periodically as the SS design and operations concepts as well as the SS MPS concept evolve.

1.2 SCOPE

The major segments of this plan are as follows:

Section 2, Project Technical Description. Includes an overview of the SS MPS and a description of its required capabilities including the computer programs identified as configurable items with an explanation of the place and function of each within the system.

Section 3, Project Activities. Presents an overview of the project plan and a detailed description of each development project activity breaking each into lower level tasks where applicable.

Section 4, Development Schedules and Manpower Requirements. Identifies the resources required and recommendations for the manner in which they should be utilized including recommended schedules and estimated manpower requirements.

Section 5, Software Development Procedures. Describes the practices, standards and techniques recommended for SS MPS Software (SW) development.

Section 2

PROJECT TECHNICAL DESCRIPTION

The SS Payload Mission Planning System (MPS) is a computer-based system that aids SS users and mission planning personnel in developing payload on-orbit operations plans and schedules. The MPS is a modularized system that encompasses planning functions from initial user operations requirements definition to generation of executable plans and real-time replanning. The MPS SW functional requirements are derived from the SS MPS Functional Flow Concept presented and discussed in Volume II of this report. The scope of the SS payloads to be scheduled include all payload operations included within or attached to the SS manned base.

A considerable portion of the SW to be utilized in the SS MPS was previously developed and utilized in the Spacelab (SL) Payload Mission Integration Planning System (MIPS) for preflight planning and real-time replanning for Spacelab payloads. Because of the similarity in some functions between the SS MPS and the SL MIPS, it was determined to be more cost effective to modify the SW corresponding to these functions for use in the SS MPS than to generate totally new SW.

A hierarchical depiction of the computer programs included in the SS MPS is presented in Figure 2-1. For the purposes of this plan it is assumed that each of the computer programs identified on this figure are classified as configurable items. The exception to this is the "Data Flow" block which in reality will encompass various interrelated data flow analysis and planning computer programs. The actual functional breakdown of the SS data communications system is presently inadequate for the purposes of function allocation to computer programs.

The decision to develop the SS MPS in the architecture described -- loosely coupled, interrelated computer programs -- evolved based on several factors:

(1) The present architecture of the SL MIPS.
(2) The number and complexity of planning interfaces.
(3) The ability to clearly partition functions into loosely related modules.
(4) The desire for a structured modularized system that will present the maximum benefits in overall system flexibility, ability to evolve/expand and system maintenance and configuration control.

A number of the SW modules identified in Figure 2-1 are annotated as Artificial Intelligence (AI) candidates. A description of the AI techniques to be utilized in these computer programs is included in the functional descriptions of the individual programs. Additional depth on the AI considerations is included in Volume II of this report.

A functional description of each of the computer programs identified in Figure 2-1 and included in the SS MPS is presented in Appendix A of this volume.

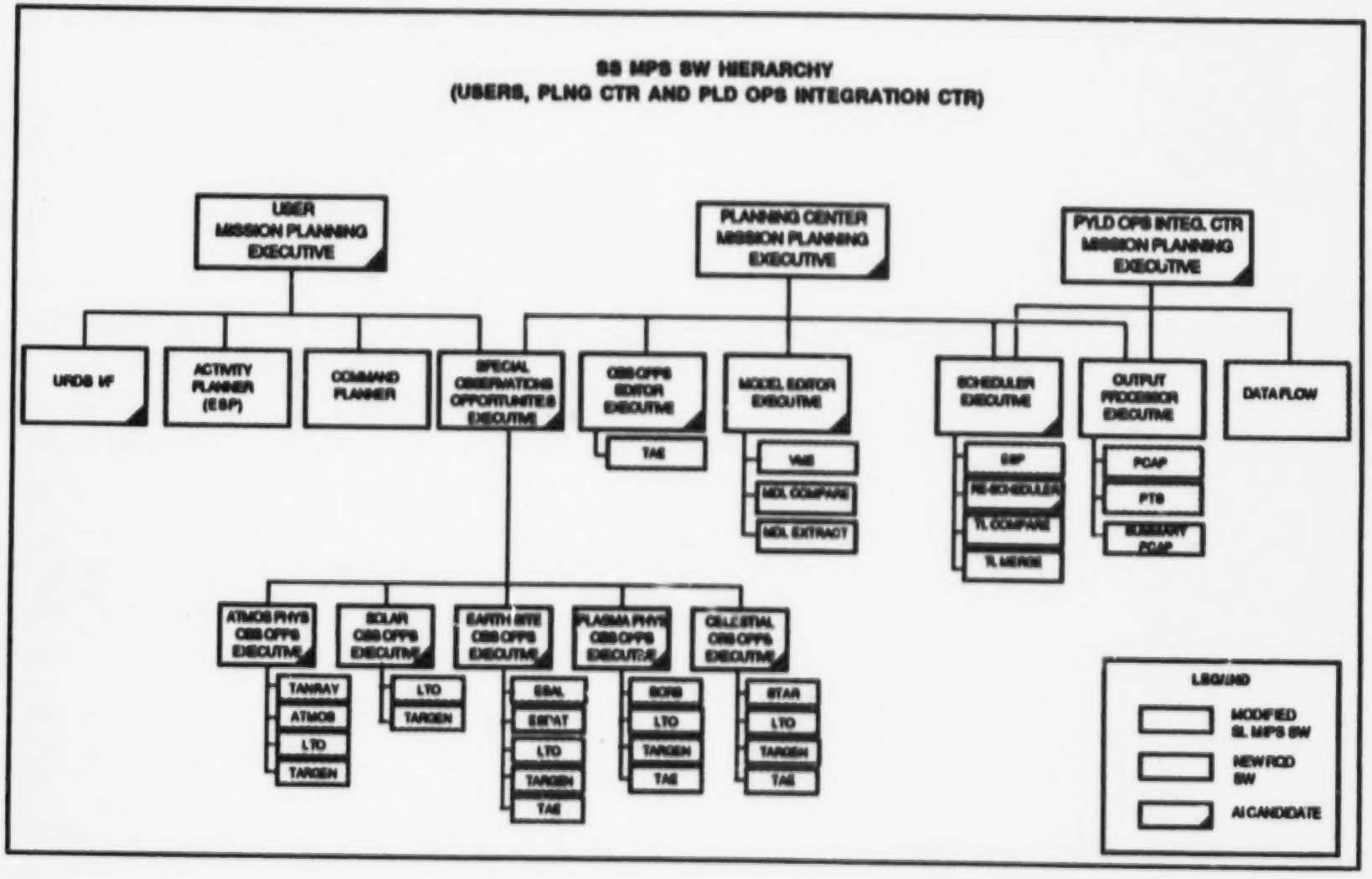


FIGURE 2-1. SS MPS SW HIERARCHY

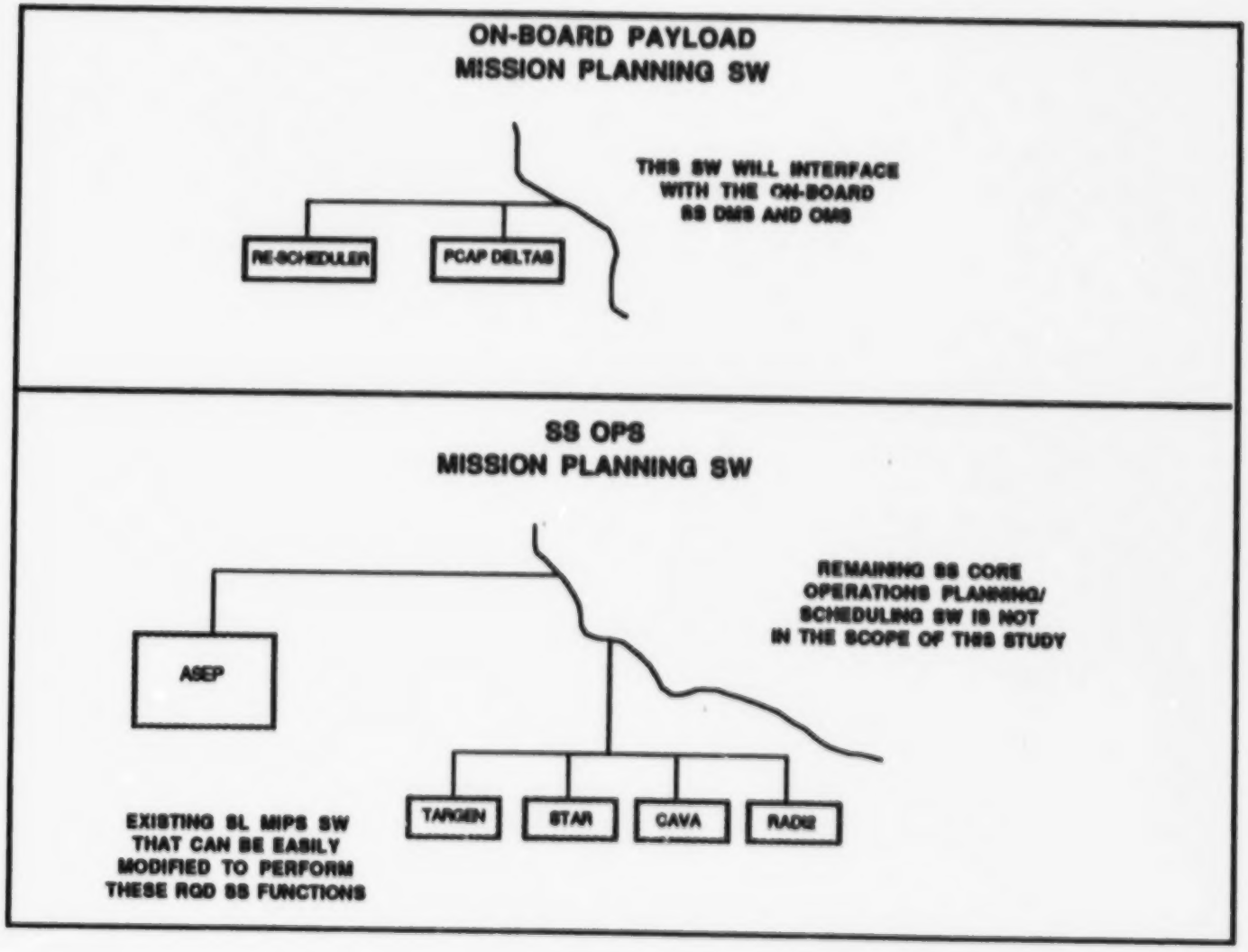

ON-BOARD PAYLOAD
MISSION PLANNING SW
THIS SW WILL INTERFACE WITH THE ON-BOARD SS DMS AND OMS
RE-SCHEDULER
PCAP DELTAS
SS OPS
MISSION PLANNING SW
REMAINING SS CORE OPERATIONS PLANNING/ SCHEDULING SW IS NOT IN THE SCOPE OF THIS STUDY
ASEP
TARGEN
STAR
CAVA
RADS
EXISTING SL MPS SW THAT CAN BE EASILY MODIFIED TO PERFORM THESE RQD SS FUNCTIONS

FIGURE 2-1. SS MPS SW HIERARCHY (CONT'D)

Section 3

PROJECT ACTIVITIES

3.1 [illegible]/IEW

3.1.1 Major Activities and Formal Reviews

The major activities of software development for the SS MPS can be divided into two groups: (1) those performed at the SS MPS system level and (2) those performed at the SW computer program (configurable item) level.

Most major activities culminate in a formal review. System level activities (and the resulting formal reviews) include:

(1) System Requirements Definition/Design (System Specification Review)
(2) System Level Simulations (System Performance Review)

Software computer program level activities (and resulting formal review) include:

(1) Software Requirements Definition (Software Requirements Review)
(2) Preliminary Program Design (Preliminary Design Review)
(3) Detailed Design and Analysis (Critical Design Review)
(4) Coding and Unit Testing (No formal review required)
(5) Module Integration and Testing (Test Procedures Review)
(6) Computer Program Testing (Software Acceptance Review)

A description of each of the major activities identified above is presented in the following sections. The SW computer program level activities described above will be performed in sequence for each computer program individually or for SW sets -- groups of similar programs at the same level in the SS MPS hierarchy that for configuration control purposes may be developed simultaneously.

3.1.2 Documentation

Table 3.1-1 identifies the formal reviews required at completion of the major activities for the SS MPS or for each computer program or SW set. The purpose of the review and the documentation to be reviewed are also shown.

Table 3.1-2 defines the documents required for successful completion of the SS MPS SW Development Project. A System Specification and System Simulation Report will be required for the overall SS MPS. The other documents listed are required for each computer program.

TABLE 3.1-1 REQUIRED FORMAL REVIEWS

NAME	PURPOSE	MATERIAL REVIEWED
SSR (System Specification Review	Verify system requirements	System Specification
SRR (Software Requirements Review)	Verify software requirements	Software Requirements Specification
PDR (Preliminary Design Review)	Verify design approach and interfaces	Functional Design Document Test Plan (preliminary)
CDR (Critical Design Review)	Verify detailed design and test plan	Detailed Design Document Test Plan Test Descriptions
TPR (Test Procedures Review)	Verify test procedures	Test Plan Test descriptions Test Procedures
SAR (Software Acceptance Review)	Verify test results and completed product (code and documentation)	Test Procedures Test Report Version Description Document Computer Systems Operators Manual Software Users Manual "As-Built" Updates to: Functional Design Document Detailed Design Document
SPR (System Performance Review)	Verify simulation results for total system	Simulation Report

TABLE 3.1-2 REQUIRED FORMAL DOCUMENTATION

DOCUMENT	PURPOSE	REVIEW	REMARKS
System Specification	Define System Requirements	SSR	
Software Requirements Specification	Establish software requirements to provide basis for design	SRR PDR	
Functional Design Document	Establish a software functional design that satisfies specified requirements	PDR CDR SAR	"As-built" update
Software Test Plan	Establish requirements, responsibilities, schedules for software testing	PDR CDR	Preliminary
Software Test Description	Describe test cases for software testing	CDR TPR	
Software Test Procedures	Establish procedures for test conduct	TPR SAR	
Software Test Report	Document the test results	SAR	
Detailed Design Document	Provide complete software design	CDR	
Computer System Operators Manual (CSOM)	Describe hardware configuration requirements for operating the software	SAR	
Software Users Manual (SUM)	Describe procedures for operating the Software	SAR	
Version Description Document	Describe content and capability of delivered Software version	SAR	

3.1.3 Definition of Computer Program Structure

Computer program structured design (Yourdon, E. and L. Constantine, Strucutred Design, Yourdon and Company, Inc., 1976) is a method of defining the architectural structure of a computer program using a top down method of design. From the top down, a "program" consists of "modules" which in turn consist of "units." These terms are used with this meaning in this plan.

3.2 SS MPS SYSTEM REQUIREMENTS DEFINITION/DESIGN

Because of the modularized nature of the SS MPS system - each program being loosely coupled to other programs or executives - a comprehensive SW system design must be performed before developing each of the independent programs. The design must be updated periodically as the development of each of the individual programs proceed. Changes in the system design resulting from computer program requirements changes must be fed back to any other impacted programs. The activities performed during this phase consist of the requirements analysis and trade studies to determine: (1) the SS MPS functions to be performed, (2) how well the functions are to be performed, (3) how the system will be structured or segmented, (4) the allocation of top level requirements to individual segments (computer programs), and (5) the definition of system-level data base requirements. The SS MPS functional flows, SW hierarchy, and SW functional requirements included in Volume II of this report should be used as the baseline for performing these tasks.

The SS MPS system design depends on several external factors, most notably the SS design and operations concepts and the interfaces with SS systems operations, users and user working groups. Because these factors will most likely be transient over the course of the SS MPS project a systematic design approach documenting open items and phasing SW computer program development based on the attainable level of requirements definition must be followed.

The output of this phase is the SS MPS System Specification documenting the results of the above analysis. A formal review of the System Specification is required to baseline the document. This document will require periodic updating (formally controlled) as the SS MPS software development proceeds.

3.3 SOFTWARE REQUIREMENTS DEFINITION

Software requirements will be analyzed to completely define the functional, performance, interface and verification requirements of each computer program. This activity extracts SW requirements from the System Specification and derives additional detailed requirements.

A description of each of the tasks comprising this activity follows:

(1) Requirements Analysis and Allocation. All system level requirements pertaining to an individual program are expanded to provide a clear definition of the functions and performance parameters for the computer program.

(2) Operational Sequence Analysis. The computer program level functional requirements contained in the System Specification are extended to a lower level of detail to describe the detailed functions of the individual program. The emphasis is to derive detailed requirements as they relate to other MPS programs.

(3) Interface Definition. This activity involves using the DeMarco/Yourdon Structured Analysis techniques including data flow diagramming, preparing a data dictionary of all interface data, and creating process/functional descriptions. Interface analysis, human factors analysis and design tasks must be performed to derive the associated software requirements. This activity will take into consideration partitioning functional requirements to minimize interfaces, providing traceability to the System Specification, and providing for completeness, consistency, and testability of the requirements.

(4) Participate in Walkthroughs and Reviews. This activity subjects the evolving requirements specifications - that is, the data flow diagrams and process descriptions - to review by managers, quality evaluation personnel, and software engineering peers. Resultant issues and decisions will be documented in walkthrough reports and internal review reports.

(5) Finalize the Software Requirements Specification (SRS). This activity involves documenting the results of the previous analysis and review activities. A separate SRS will be developed for each of the computer programs and each SRS will contain the Interface Requirements Specification (IRS) for the computer program. The SRS includes textual and graphical descriptions of interface identifications and summaries, interface data, function inputs and outputs, requirements traceability, and qualification methods.

(6) Participate in the Software Specification Review (SSR). This activity involves preparing for and participating in the SSR. The data item to be reviewed is the SRS.

3.4 PRELIMINARY PROGRAM DESIGN

Preliminary program design is the process of defining the overall structural design at the computer program level. This process includes the allocation of functions to lower level program modules, definition of the interfaces between these modules, development of the data base concept and development of a verification plan.

A description of each of the tasks comprising this activity follows:

(1) Preliminary SW Design. Computer program functions specified in the SRS are organized into modules, the first structural level below the program. Program structure and operating concepts will be defined in this phase and should require little variation in subsequent phases. Actual module design could change significantly in the detailed design and analysis phase. The operational concept of each program will be defined, specifying how the program will be controlled to accomplish its function. The concept of control, sequencing of executable elements, interrupt handling, and input/output handling are developed at this point. Operational modes are defined and operational timelines developed which describe the expected sequence and timing of executable elements for normal and abnormal conditions.

(2) Timing and Sizing Studies. Estimates of the time required to execute each executable element and the amount of memory required during execution will be prepared. These estimates, particularly that of time, will be of importance to planned use of executable elements in the real-time mission replanning cycle.

(3) Preliminary Interface Design. Interfaces between the computer program and external sources will be defined by types of data, data rates, special conditions, interface protocol and special data items. Because of the phased nature of the MPS software development this task will be somewhat fragmented. Computer programs that are developed first will assume interfaces with programs to be developed downstream. System level design will alleviate the problem by supplying as much detail as possible at the top level.

(4) Data Base Design. The structure of the data base, access methods, updating methods, control and protection procedures will be determined. The phased SW development will again impact this task. Data bases that are common between computer programs developed out of phase will most likely require iterations to complete their design. System level data base requirements will be as detailed as possible to alleviate this problem.

(5) Participate In Walkthroughs And Reviews. This activity subjects the evolving FDD and operation and support documentation to review by managers, quality evaluation personnel, and software engineering peers. Resultant issues and decisions will be documented in walkthrough reports and internal review reports.

(6) Establish the Software Development Library (SDL). This activity begins with the entry of the customer-approved specification (SRS) into the SDL. The SDL will contain documentation, tools, and evolving software needed during the design, coding, and testing of the software. (See paragraph 5.2.)

(7) Finalize Functional Design Document (FDD). This activity incorporates the results of the design and review activities above to produce a deliverable FDD.

(8) Test Planning. A preliminary Software Test Plan (STP) for each computer program will be developed.

(9) Participate In Preliminary Design Review (PDR). This activity involves preparing for and participating in the software PDR, which is a formal review. The data items to be reviewed are the FDD and preliminary STP.

3.5 DETAILED DESIGN AND ANALYSIS

The technical objectives for this phase are to complete and review the design of each computer program, and the computer program interfaces.

A description of each of the tasks comprising this activity follows:

(1) Analysis. This activity includes equation or algorithm derivation, data base content analysis, data storage and access analysis, throughput analysis and software design analysis.

(2) Interface Design. Interfaces designed during preliminary design will be finalized and changes coordinated on both sides of the interface.

(3) Software Modeling. This activity consists of experimental coding of the functions to be evaluated and dummy representations of the interfacing elements not required to verify the design solutions.

(4) Software Operation Design. Includes support of initiation and operation of the computer program. Initiation involves activating the software, controlling its execution and operating backup configurations. Operation design involves the detailed definition of all interactions with human users of the software, including display formats, command inputs, operational restrictions and error conditions.

(5) Detailed SW Design. Develop a detailed logic flow for all levels of SW in a top down manner.

(6) Create Software Development Files (SDFs). SDFs will be created to correspond to all units now defined as a result of the decomposition of the higher level elements. Each SDF will correspond to a unit or logically related group of units. (See paragraph 5.3.)

(7) Develop A Software Test Description (STD). This activity will produce an STD for each computer program to describe the test cases for each formal test. Descriptions include initialization information, input data, intermediate test results, output data, and criteria for evaluating results. These will be submitted as a part of the CDR data package, plus a finalized version of the STP.

(8) Prepare Documentation For Unit Test Cases And Integration Test Cases. This activity will identify the requirements, responsibilities, and schedule. General information had previously been included in the STP. Since module and unit identifications will be available at the time of this activity, details specific to each element can be documented. This activity will also describe inputs, expected results, and evaluation criteria for the informal test cases.

(9) Participate in Walkthrough and Reviews. This activity subjects the evolving detailed design to review by managers, quality evaluation personnel, and software engineering peers. Allocation of requirements to the modules and units will be assessed. Projected sizing and timing budgets and margins will be reviewed. As in the preliminary design activity an iteration in module design may be warranted if allocated budgets and projected budgets have significant variations. The walkthroughs will assist in the detection of interface and implementation design problems. The walkthroughs will occur after a significant amount of design has been produced but early enough so that detected flaws can be corrected before major rework is required.

(10) Prepare the Software Detailed Design Document (SDDD). This activity incorporates the results of the design and review activities above to produce a deliverable SDDD for each computer program. Included in each SDDD will be the Interface Design Document (IDD) and Data Base Design Document (DBDD). The SDDDs will be submitted as a part of the CDR data package.

(11) Participate in the Critical Design Review. This activity involves preparing for and participating in the CDR, which is a formal review. The data items to be reviewed are the SDDD, STP, and STD.

3.6 CODING AND UNIT TESTING

Units will be coded in a top-down manner and each will be tested for verification of its processing, data manipulations, and error handling.

A description of each of the tasks comprising this activity follows:

(1) Code Units in a Top-Down Manner. Each unit will be coded per the coding standards specified in paragraph 5.5. Some coding standards will be unique to a computer program. This uniqueness will be due to pecularities of each of the programs, software development facilities, the System Requirements and the run time operating systems. Any exceptions made to top-down coding will be made to address critical units.

(2) Prepare Unit Test Procedures. This activity will define the test procedures for unit testing including the objectives of the test case, the test methods, inputs, and expected outputs.

(3) Perform Unit Tests. This activity will involve obtaining error-free compilations, debugging in-line to allow static analysis and breakpointing, stubbing of called units, and some integrating of modules to ensure consistent testing.

(4) Update the SDFs. This activity involves recording the results of the unit tests and inserting these test results and the current source code listings into the corresponding SDFs.

(5) Prepare Module Level Integration Test Procedures. This activity produces documentation for the Module Level Integration Test Procedures including objectives, test methods, inputs, and expected outputs of the testing.

(6) Prepare A Preliminary Software Test Procedure (STPR) For Each Computer Program. This activity documents the procedures to be used for formal testing, that is, for computer program testing. The documentation includes pretest procedures, step-by-step procedures, and the procedures to be used for data reduction and data analysis.

(7) Participate In Walkthroughs and Reviews. This activity will subject the unit with its hardcopy code listing and test results to review by managers, software quality evaluation personnel and software engineering peers. This activity will assist in detecting interface and flow problems, inconsistencies with coding standards, and deficiencies in testing. Any issues or decisions that result from the review will be recorded in the SDF.

(8) Develop Operation and Support Documentation. This activity will produce the Software User's Manual (SUM) and Computer System Operator's Manual (CSOM) for each computer program.

3.7 MODULE INTEGRATION AND TESTING

Units will be integrated into modules tested in accordance with the module integration test cases and Module Integration Test Procedures.

A description of each of the tasks comprising this activity follows:

(1) Integrate Units Into Modules. Units will be integrated to form higher level elements so that testing of the aggregates may be performed. The units will also be integrated to form the computer programs; however, testing of the computer programs will occur during the next major activity.

(2) Test Modules. Tests will be performed on the modules according to the documented test cases and test procedures. The testing will produce hardcopy outputs onto which annotations will be marked to show where the objectives and requirements have been met. Discrepancies will be documented. Recommendations for corrective action and retest will be made.

(3) Assess Memory Use and Processing Times. This activity will contrast memory and processing time allocations made during the design activities with memory and processing time values obtained when aggregates for units were tested together. Also assessed will be any required system

resources that may differ from earlier specifications. Any necessary changes will be made to the documentation to reflect the new assessments of memory and processing time requirements and any new system resource requirements.

(4) Record The Test results. Module integration test results will be recorded in the format given in paragraph 5.4.

(5) Perform Needed Corrections and Regression Testing. Corrections will be made as necessary to design documentation and code. Required regression testing will be performed. The SDFs that correspond to the units that have undergone design documentation or coding changes will be updated.

(6) Finalize The STPRs For Each Unit. The preliminary STPRs for each computer program prepared during the previous major activity will be finalized.

(7) Participate In Walkthroughs And Reviews. This activity subjects the module integration test results, computer program formal test procedures and evolving SUMs and CSOMs to review by managers, software quality evaluation personnel and software engineering peers. Resultant issues and decisions will be documented in walkthrough reports and internal review reports.

(8) Update the SUM and CSOMs. The evolving versions of the SUM and CSOM for an computer program will be updated with any known details.

(9) Participate In The Test Procedures Review (TPR). This activity involves preparing for and participating in the TPR, which is a formal review. The data items to be reviewed are the module integration tests results, STPRs for each computer program and the evolving SUM and CSOM for each computer program.

3.8 COMPUTER PROGRAM TESTING

The computer programs are tested in accordance with formal test documentation. Then all software and documentation is readied for audit and delivery or baselining for use.

A description of each of the tasks comprising the activity follows:

(1) Test Computer Programs. This activity involves testing each program in accordance with the formal test documentation which includes the STP, STD, and STPR. Testing will be performed by individuals who are independent from the developers.

(2) Prepare SW Test Report (STR). Record the formal test results and prepare a Software Test Report (STR) containing a summary of the tests, test history, results of each formal test, test result evaluations and recommendations, and deviations. This test reporting will be performed by individuals independent from the developers.

(3) Perform Corrections and Retest. Corrections will be made as necessary to design documentation and code. Required retesting will be performed.

(4) Prepare a Version Description Document (VDD) for each Computer Program. This activity involves identifying the exact version of each computer program and the interim changes that occur between versions. The VDD is a living document intended to maintain a change history for the computer program and allow users of the software to identify changes made between different releases.

(5) Finalize The SUM and CSOM For Each Computer Program. The completed versions of the SUM and CSOM for each computer program will be prepared.

(6) Participate in Walkthroughs and Reviews. This activity will subject the completed software and documentation to review by managers, software quality evaluation personnel, and software engineering peers.

(7) Participate in Software Acceptance Review (SAR). Formal test results in the STR will be reviewed to verify that the computer program was successfully tested. A check will be made to see if the requirements of the SRS have been met. The VDD will be reviewed to make sure it reflects an accurate technical description of the computer program. The final versions of the SUM and CSOM will be evaluated with respect to how well they address operation and support of the computer system.

(8) Prepare Computer Program For Delivery. The deliverable versions of source and object code for each computer program will be prepared for delivery in accordance with the requirements stated in the SRS for the computer program.

3.9 UNIQUE ACTIVITIES IN SS MPS DEVELOPMENT (COMPUTER PROGRAM LEVEL)

The following activities are unique to one or more of the SS MPS computer programs.

3.9.1 Prototyping

This activity is performed prior to Software Requirements Specification. Prototyping provides working systems/subsystems at a gross operational level to support implementation feasibility. The end product of prototyping is a set of well defined requirements. For several software modules in the MPS system, the cost of prototyping is likely to be much less than the cost associated with requirements redefinition late in the development cycle.

Several software modules have been recommended for prototyping in the LISP language/Symbolics machine environment. This environment affords more operator flexibility and time savings than conventional hardware/software environments.

3.10 SS MPS SIMULATIONS AND PERFORMANCE REVIEW

After each computer program has been acceptance tested according to formal test documentation, overall system simulations will be conducted to verify the performance of the SS MPS. Simulations will consist of various payload complement test cases. At the completion of the simulations a SS MPS performance review will be held.

A description of each of the tasks comprising this activity follows:

(1) Development Simulation Plans. This activity produces a Simulation Plan which identifies the requirements responsibilities, and schedules for the Simulations.

(2) Develop A Simulation Description. This activity will produce a description to describe the payload complement test cases. Descriptions include input data, intermediate results, output data and criteria for evaluating results.

(3) Perform SS MPS Simulations. This activity consists of performing a Simulation of the overall SS MPS according to the documented Simulation Description. The Simulation will produce hardcopy outputs onto which annotations will be marked to show where the objectives and requirements have been met. Discrepancies will be documented. Recommendations for corrective action and retest will be made.

(4) Document Results. The results of the Simulation will be documented in a Simulation Report to be reviewed at the System Level Performance review.

(5) Participate in Performance Review. Simulation results will be reviewed to verify that the MPS performed successfully. Open items will be documented as will any required iterations in the MPS development.

Section 4

DEVELOPMENT SCHEDULES AND MANPOWER REQUIREMENTS

The estimated cost of the SS MPS project in terms of manpower and schedule is presented in this section. The approach to cost estimating was to group the computer programs identified in Section 2 into the SW Sets defined in Table 4-1 based on similarities in computer program type, interfaces, AI recommendations and SW hierarchy. The cost of each of these sets was estimated by use of the Constructive Cost Model (COCOMO). The effort required for prototyping, system level design and system level testing was estimated separately.

4.1 METHODOLOGY AND ASSUMPTIONS

The COCOMO estimating procedure is driven by program size (lines of code) and various cost drivers as described in the following paragraphs.

4.1.1 Lines Of Code

The lines of code estimates for each new computer program (see Table 4-1) were arrived at by allocating functions to each computer program (see Appendix A), estimating the lines of code for each function and summing these to arrive at the total computer program estimate.

The original lines of code for the modified SL MIPS computer programs were taken from the SL MIPS Data Base presented in Volume II, Appendix A. Design, coding and integration modification factors were estimated subjectively based on the number of functions to be retained from the reused code and the number of new functions to be added.

4.1.2 Software Cost Drivers

The various cost drivers employed by COCOMO are shown in Figure 4.1.2-1 along with the ratings assumed for each of the SS MPS SW Sets. Definitions of each of the cost drivers and how they are applied in the cost estimating process can be found in Software Engineering Economics, Barry Boehm; Prentice-Hall, Inc.; Englewood Cliffs, N. J.; 1981.

4.2 MANPOWER REQUIREMENTS

The manpower requirements are an output of COCOMO for each SW Set development effort with manual adjustments made for prototyping, system level design and system level simulations and performance reviews.

4.2.1 SW Set Manpower Requirements

The number of manmonths per SW Set Phase required to successfully complete the SS MPS project is illustrated for the individual SW Sets in Figure 4.2.1-1. An activity distribution of manpower corresponding to this phase distribution is included in Appendix B.

TABLE 4-1 SW SET GROUPINGS AND LINES OF CODE ESTIMATES

NEW SOFTWARE	ESTIMATED LINES OF CODE MODULE	SET
SYSTEM EXECUTIVES (PHASE I)		37,000
USER MPS EXEC	9,000	
PLANNING CENTER MPS EXEC	14,000	
POIC MPS EXEC	14,000	
SYSTEM EXECUTIVES (PHASE II)		55,000
USER MPS EXEC	15,000	
PLANNING CENTER MPS EXEC	20,000	
POIC MPS EXEC	20,000	
SPECIAL OBS OPPS EXECUTIVES		42,000
TOP LEVEL	22,000	
ATMOS PHYS	4,000	
SOLAR	4,000	
EARTH SITE	4,000	
PLASMA PHYSICS	4,000	
CELESTIAL	4,000	
EDITOR EXECUTIVES		18,000
MODEL EDITOR EXEC	6,000	
OBS OPPS EDITOR EXEC	7,000	
SCHEDULER EXEC	5,000	
RE-SCHEDULER		36,000
URDB I/F		41,000
COMMAND PLANNER		10,000
OUTPUT PROCESSOR EXEC		8,000
NEW TIMELINE ANALYSIS SOFTWARE		53,000
MDL EXTRACT	4,000	
MDL COMPARE	4,000	
TL COMPARE	6,000	
TL MERGE	6,000	
PCAP DELTAS	8,000	
SUMMARY PCAP	25,000	
TOTAL		300,000

TABLE 4-1. SM SET GROUPINGS AND LINES OF CODE ESTIMATES (CONT'D)

MODIFIED SL MIPS MODULES	LINES OF CODE MODULE	LINES OF CODE SET	ESTIMATED EQUIVALENT NEW LINES
ORBIT ANALYSIS		63,209	20,525
ASEP	14,191		
ATMOS	1,650		
BORB	2,100		
CAVA	20,072		
ESAL	1,850		
ESDAT	700		
LTO	351		
RADI2	8,060		
STAR	2,054		
TANRAY	1,625		
TARGEN	10,556		
TIMELINE ANALYSIS		173,200	61,395
ESP	90,000		
PCAP	27,000		
PTS	6,200		
TAE	10,000		
VME	40,000		
DATA FLOW ANALYSIS		164,224	75,461
PROFILE	5,031		
MISSION WINDOWS	16,608		
ONBOARD RECORDER SCHEDULAR	15,587		
POSSIBLE FORMATS	2,630		
FORMAT SCHEDULAR	7,908		
POSSIBLE POCC CONFIGURATIONS	9,396		
POCC CONFIGURATION SCHEDULAR	5,555		
PLAYBACK SCHEDULAR	16,498		
INTERACTIVE DATA UPDATE SYSTEM	31,790		
VERIFICATION	8,186		
COMPARE TDRS	580		
COMPARE MODELS	2,072		
DATA MANAGEMENT CHECKLIST	5,857		
DATA SCHEDULE FILE	29,412		
ANTENNA DISPLAY	4,056		
IDMS LIBRARY	3,058		
TOTAL		400,633	157,381

Because of the uncertainties of the values of the various cost drivers used in the SS MPS cost estimating procedure an analysis was performed to identify the sensitivity of the estimated outputs to variance in the assumed cost drivers. A representative SW Set, the Re-Scheduler, was chosen as an example. Five of the cost drivers were analyzed and the change in manpower corresponding to changes in these drivers is presented in Figure 4.2.1-2. The baseline assumptions are the same as shown in Figure 4.1.2-1 except for the particular driver being analyzed.

From the results it is obvious that the overall estimate accuracy relies heavily on the accuracy of the input cost drivers, and the SW cost estimates must be kept current as more definition of the project is attained.

4.2.2 Manpower Requirements Summary

A top level summary of the manpower required to successfully complete the SS MPS project is presented in Table 4.2.2-1. The summary includes the COCOMO outputs of Appendix B adjusted for prototyping and system level activities. The total estimated manpower requirement is 4841 manmonths.

An estimate ran under the same assumptions and excluding the benefit of the SL MIPS software yielded an estimated manpower requirement of 9693 manmonths.

SYSTEM MODE: EMBEDDED

	RANGE	A	B	C	D	E	F	G	H	I	J	K	L
REQUIRED SW RELIABILITY	0 TO 4	3	HIGH FINANCIAL LOSS →										
DATA BASE SIZE	1 TO 4	2	4	3	4	2	2	4	4	4	4	4	2
SW COMPLEXITY	0 TO 5	4	4	4	4	4	4	3	3	3	3	3	4
EXECUTION TIME CONSTRAINT	2 TO 5	2											
STORAGE CONSTRAINT	2 TO 5	2	ASSUMPTION IS MADE THAT SUFFICIENT DEVELOPMENT HARDWARE IS AVAILABLE SO AS NOT TO IMPACT SW DEVELOPMENT →										
VIRTUAL MACHINE VOLATILITY	1 TO 4	1											
COMPUTER TURNAROUND TIME	1 TO 4	1											
ANALYST CAPABILITY	0 TO 4	3	ASSUMED MED TO HIGH BECAUSE OF THE SL MIPS EXPERIENCE AVAILABLE TO DRAW FROM →										
APPLICATION EXPERIENCE	0 TO 4	3											
PROGRAMMER CAPABILITY	0 TO 4	2	ASSUMED NOMINAL- MIX OF EXPERIENCE AND SKILL LEVELS →										
VIRTUAL MACHINE EXPERIENCE	0 TO 3	1.5											
PROGRAMMING LANGUAGE EXPERIENCE	0 TO 3	1.5											
MODERN PROGRAMMING PRACTICES	0 TO 4	3	ASSUMED USING ADA ON VAX- ABOVE AVERAGE TOOLS AVAILABLE →										
USE OF SW TOOLS	0 TO 4	3											
REQUIRED DEVELOPMENT SCHEDULE	0 TO 4	2	AMOUNT OF SCHEDULE COMPRESSION IS UNDETERMINED AT THIS POINT →										
REQUIREMENTS VOLATILITY	1 TO 5	5	ASSUMED HIGH BECAUSE OF PHASED DEVELOPMENT OF INDIVIDUAL PROGRAMS AND OVERALL SS RQMTS UNCERTAINTIES										

A-SPECIAL OBS OPPS EXECS
B-URDB I/F
C- EDITOR EXECS (AI)
D- RE-SCHEDULER
E- SYSTEM EXECS (PHASE I)
F- SYSTEM EXECS (PHASE II)
G- USER PLANNERS
H- NEW CONV TL SW
I- MODIFIED TL SW
J- MODIFIED ORBIT SW
K- MODIFIED DATA FLOW SW
L- OUTPUT PROCESSOR EXEC

FIGURE 4.1.2-1. SOFTWARE COST DRIVER RATINGS (COCOMO INPUT PARAMETERS)

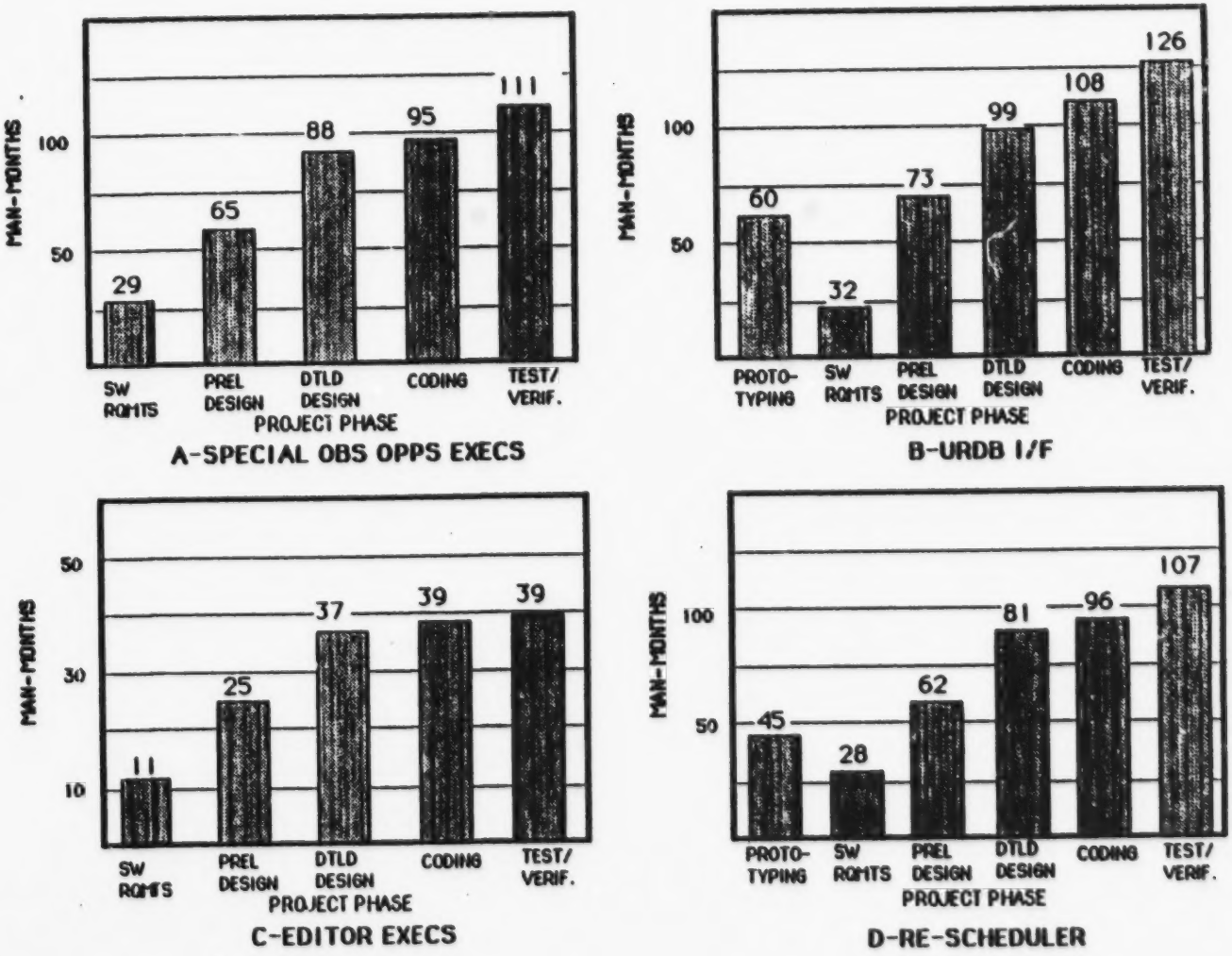


FIGURE 4.2.1-1. MANPOWER PER SW SET PHASE

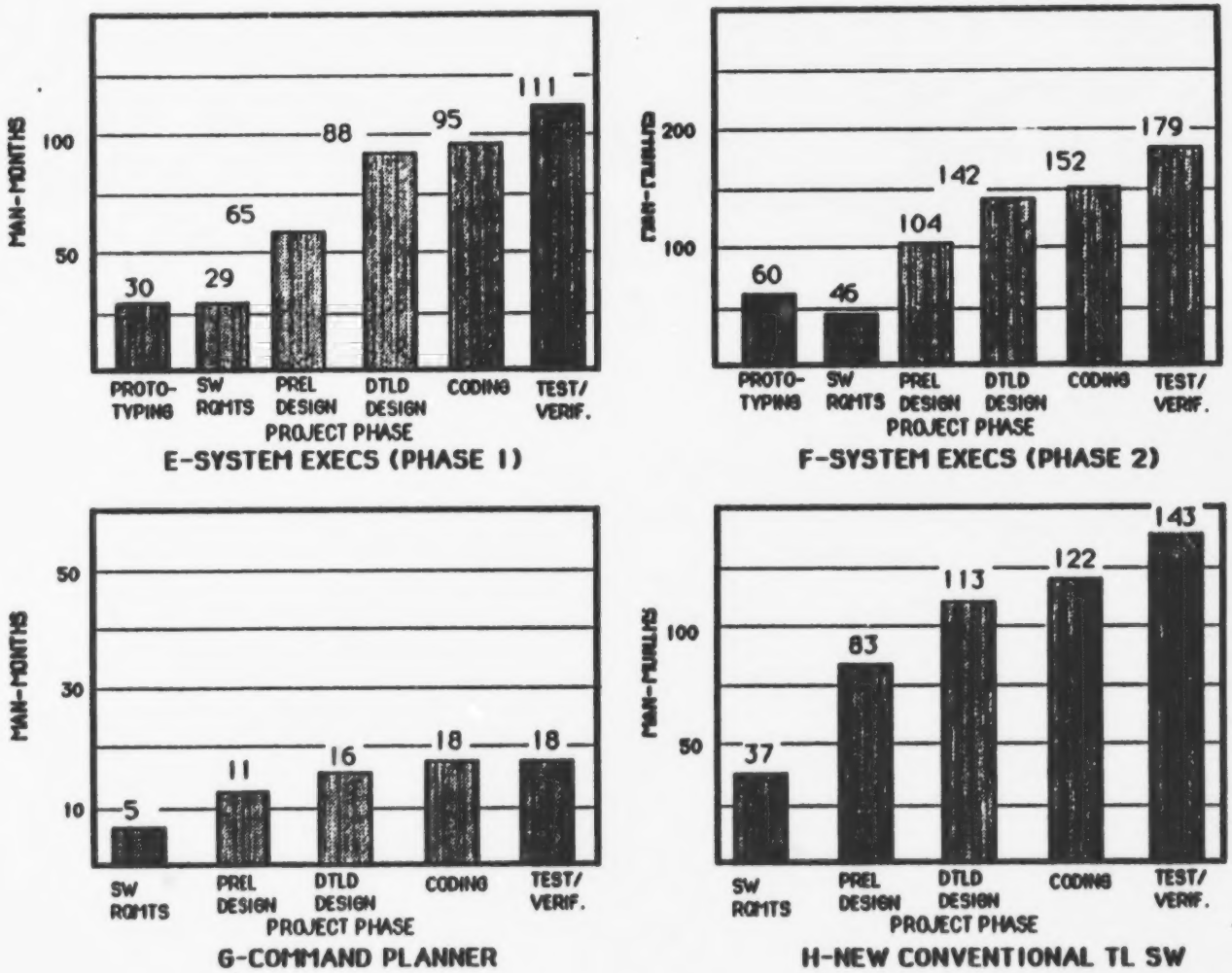


FIGURE 4.2.1-1. MANPOWER PER SW SET PHASE (CONT'D)

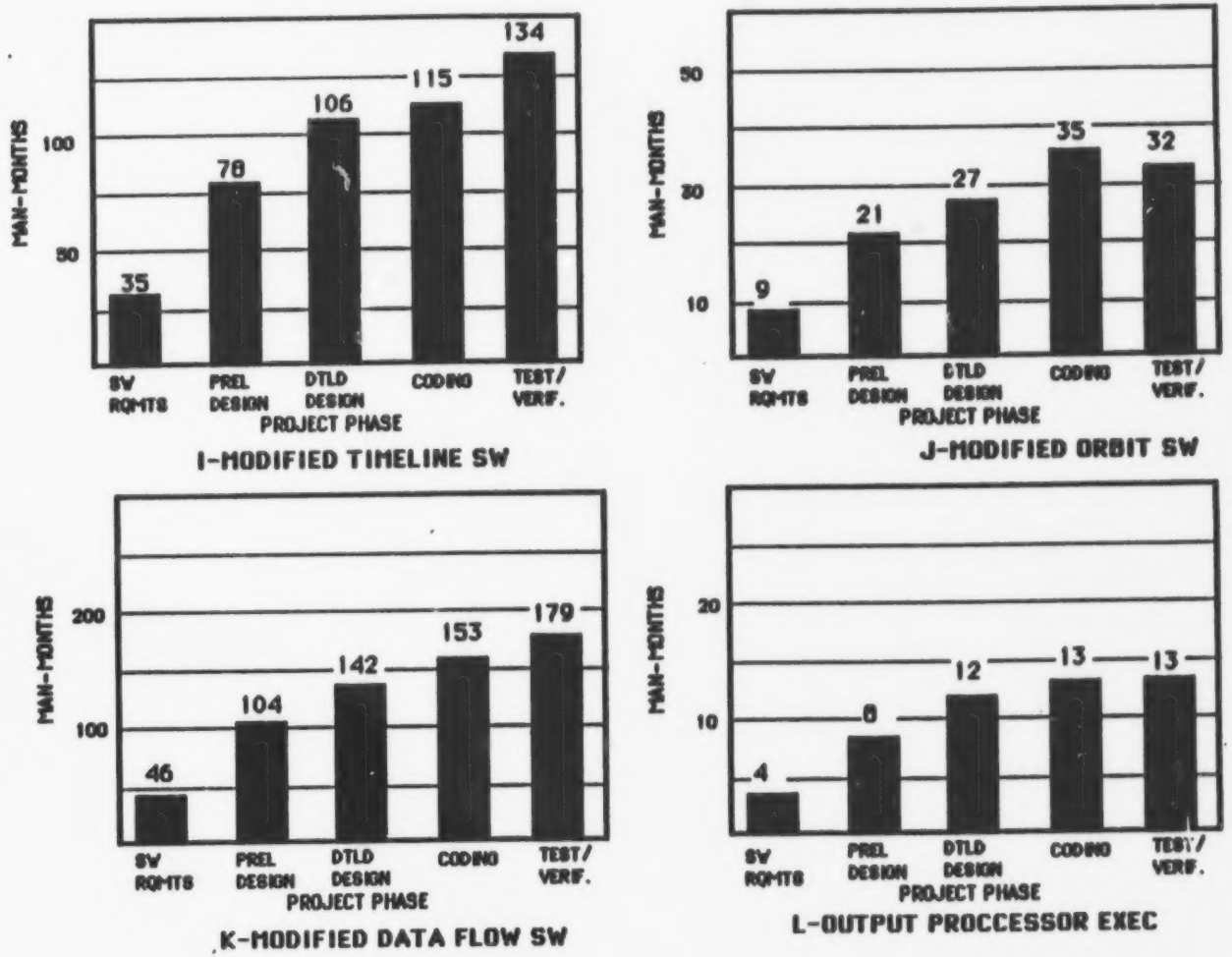


FIGURE 4.2.1-1. MANPOWER PER SW SET PHASE (CONT'D)

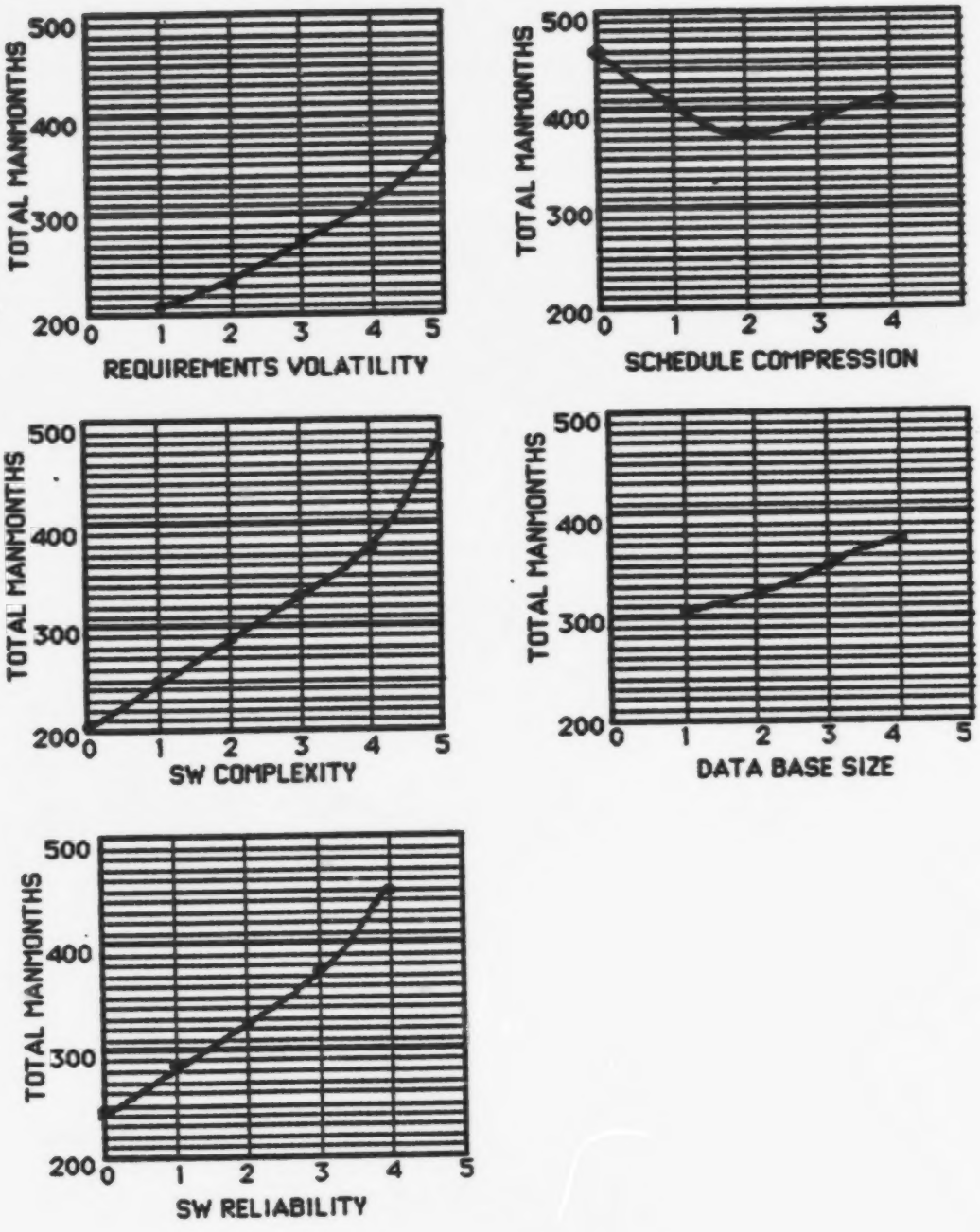


FIGURE 4.2.1-2. SENSITIVITY OF ESTIMATED MANMONTHS TO VARIOUS COST DRIVER (RE-SCHEDULER)

TABLE 4.2.2-1. MANPOWER REQUIREMENTS SUMMARY

ACTIVITIES	DURATION (MONTHS)	MANPOWER (MANMONTHS)	MAN-LOADING
MPS SYSTEM DESIGN	8	96	12
MPS SYSTEM SIMULATIONS/ PERFORMANCE REVIEW	15	360	24
SW SETS DEVELOPMENT			
A - SPECIAL OBS OPPS EXECS	28	388	13.9
B - URDB I/F	53	498	9.4
C - EDITOR EXECS	21	152	7.2
D - RESCHEDULER	46	419	9.1
E - SYSTEM EXECS (PHASE I)	40	416	10.4
F - SYSTEM EXECS (PHASE II)	57	682	12.0
G - COMMAND PLANNER	16	67	4.2
H - NEW CONVENTIONAL TL SW	31	498	16.1
I - MODIFIED TIMELINE SW	30	467	15.6
J - MODIFIED ORBIT SW	20	123	6.2
K - MODIFIED DATA FLOW SW	33	625	18.9
L - OUTPUT PROCESSOR EXEC	15	50	3.3
		4841	

4.3 SCHEDULES

4.3.1 SW Set Schedules

An activity schedule for each SW Set project phase is included in this section. The schedules are presented separately and include only delta times. The actual phasing of the SW Set development is arrived at subjectively and is discussed in Section 4.3.2.

The SW Set schedules assume that all the computer programs within a particular set all work toward the same project milestones SRR, PDR, CDR, etc. Further granularity can be obtained by scheduling activities to the computer program level, if desired, as better definition of the project is obtained. Figure 4.3.1-1 depicts the schedules for the twelve SW sets.

In determining these schedules the COCOMO output included in Appendix B was used as a basis. Subjective adjustments were made to the COCOMO output based on the estimators experience. In particular, the duration of the COCOMO programming phase output, which includes detailed design, analysis, coding, and unit testing as shown in Figure 4.3.1-1, was arbitrarily lengthened to reduce peak manpower levels.

4.3.2 Top Level SS MPS Schedule

The recommended SS MPS Top Level Development schedule is shown in Figure 4.3.2-1. This schedule was arrived at subjectively based on the evaluation of each SW Set against several factors and considering the overall SSP milestones. Figure 4.3.2-2 presents the criteria that were considered and the rating of each set versus that criteria. Obviously development lead time for the four SW Sets that require prototyping was the overriding factor and requires that development of these SW Sets begin as soon as possible. These SW Sets have sufficient requirements definition to begin prototyping concurrent with the SS MPS system level design activity. It is also recommended to begin development of A-Special Obs Opps Executives, J-Modified SL MIPS Orbit Analysis SW and I-Modified SL MIPS Timeline SW as soon as the system design activity is completed because of three factors: (1) the initial manloading of the SW sets to be prototyped is relatively low, (2) requirements definition for the modified SW should be relatively high and (3) the dependency on SS operations and design concepts for the modified SW is low to moderate. The remaining activities should be phased in based on better definition of the SS operations and design concepts and MPS concept. As these become more firm, better decisions on the actual phasing of the SW development can be made. For budgetary purposes an estimate of the required manloading for the first two years of the project was made and is presented in Table 4.3.2-1, based on the recommended top level schedule in Figure 4.3.2-1. No attempt was made to provide a distribution of manloading for the remainder of the project because of the instabilities of the long term schedule. If the project proceeds as shown on the top level schedule the manloading for the remaining 40 months of the project would average an estimated 99 men.

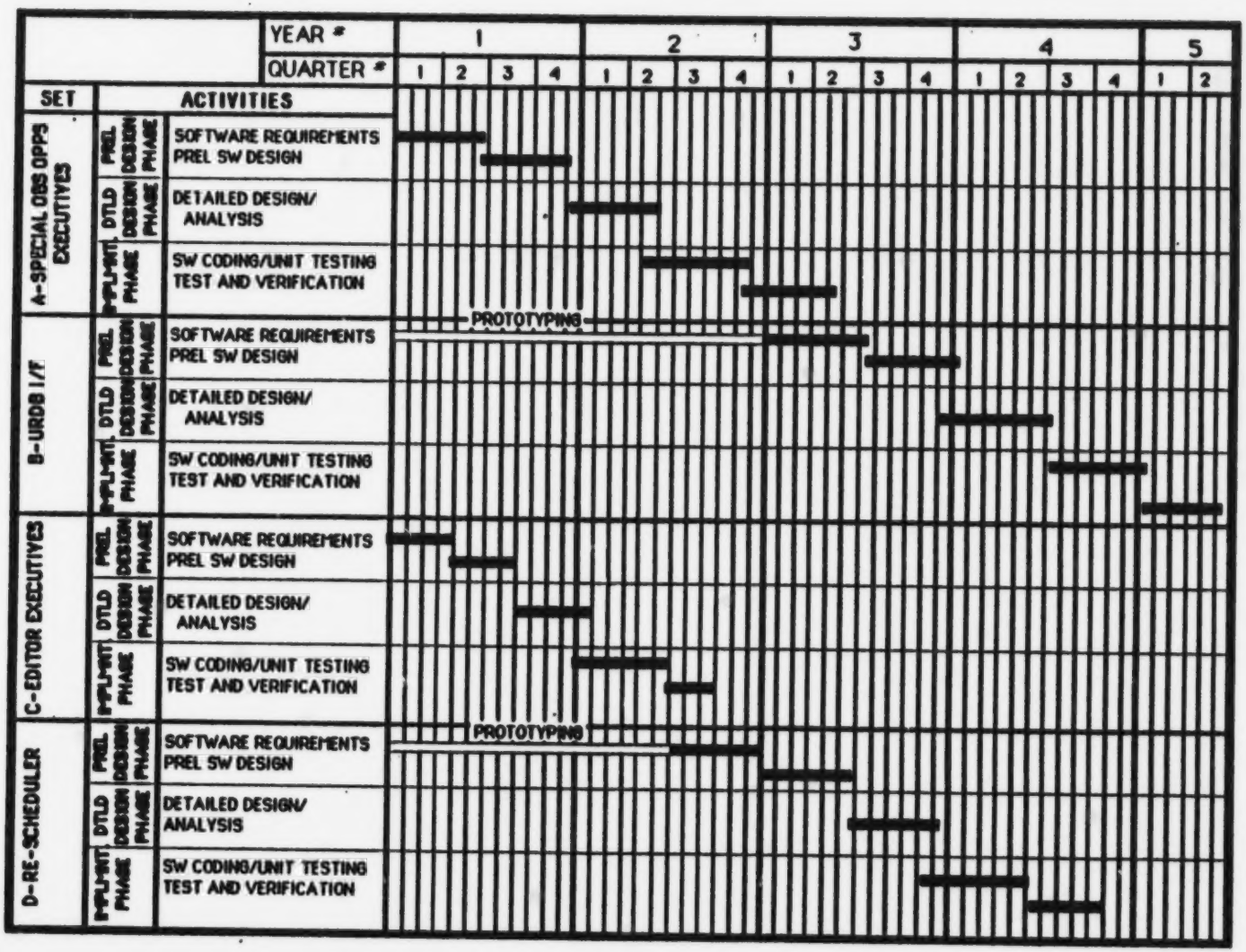


FIGURE 4.3.1-1. SW SET DEVELOPMENT SCHEDULES

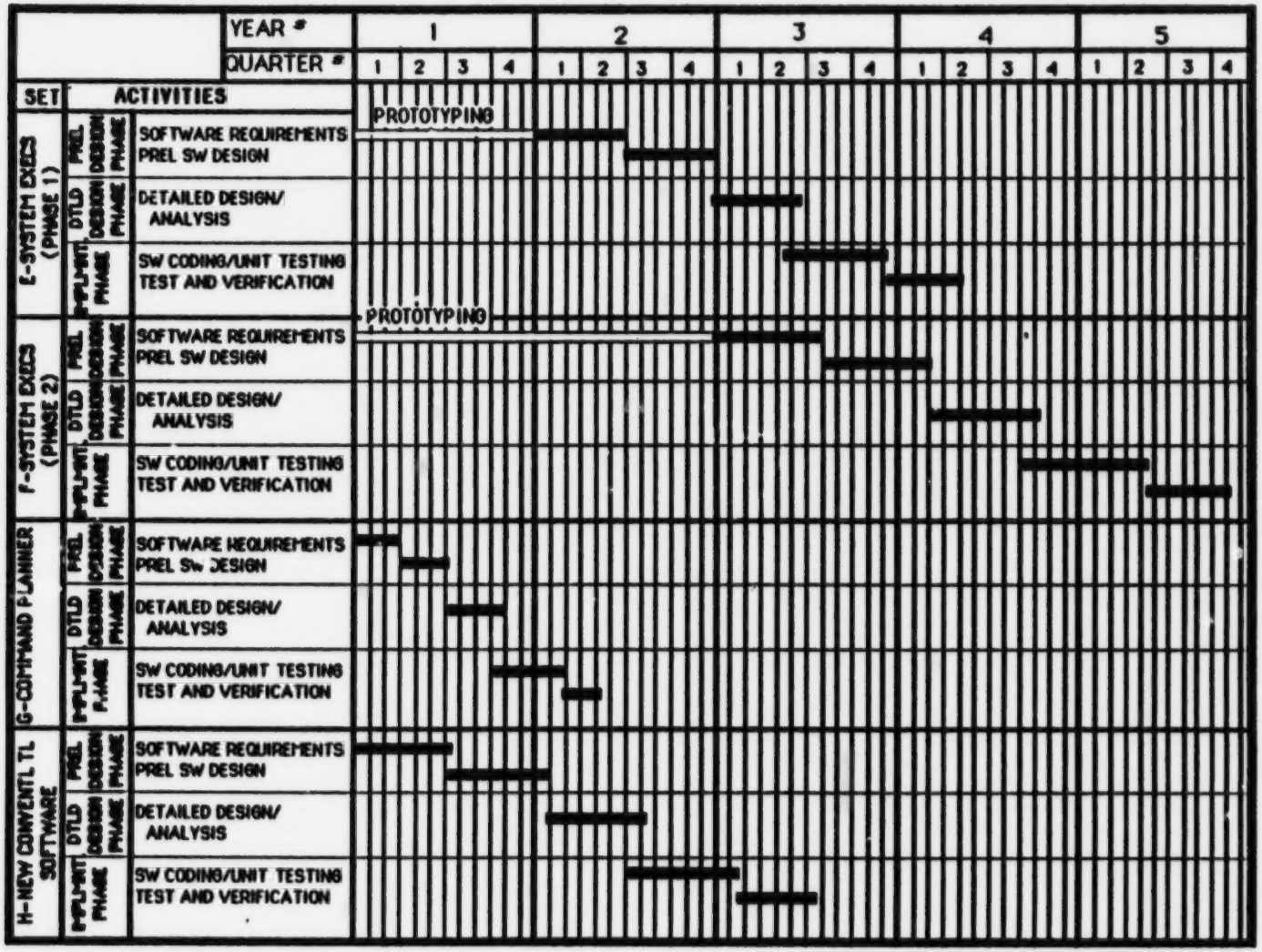


FIGURE 4.3.1-1. SW SET DEVELOPMENT SCHEDULES (CONT'D)

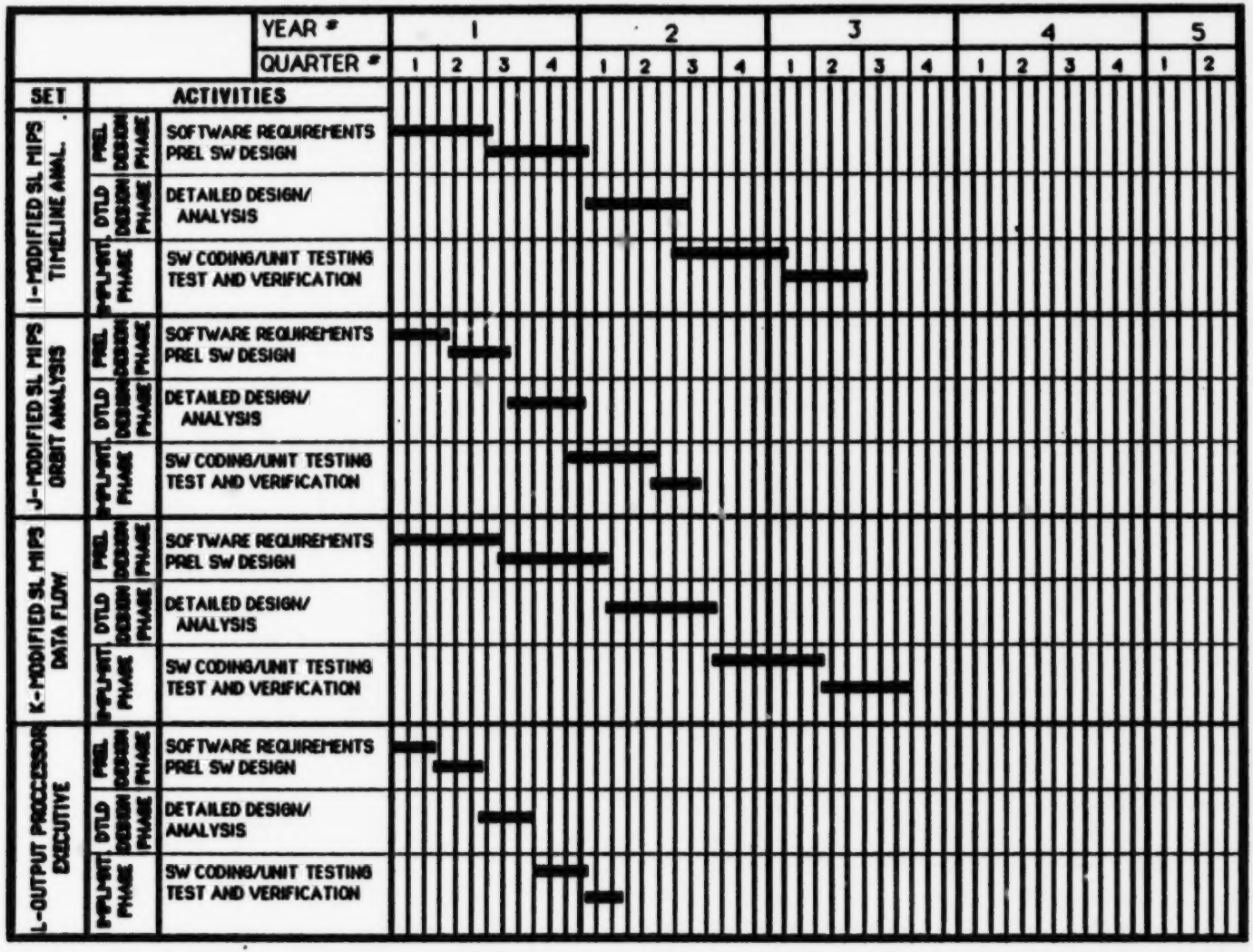


FIGURE 4.3.1-1. SW SET DEVELOPMENT SCHEDULES (CONT'D)

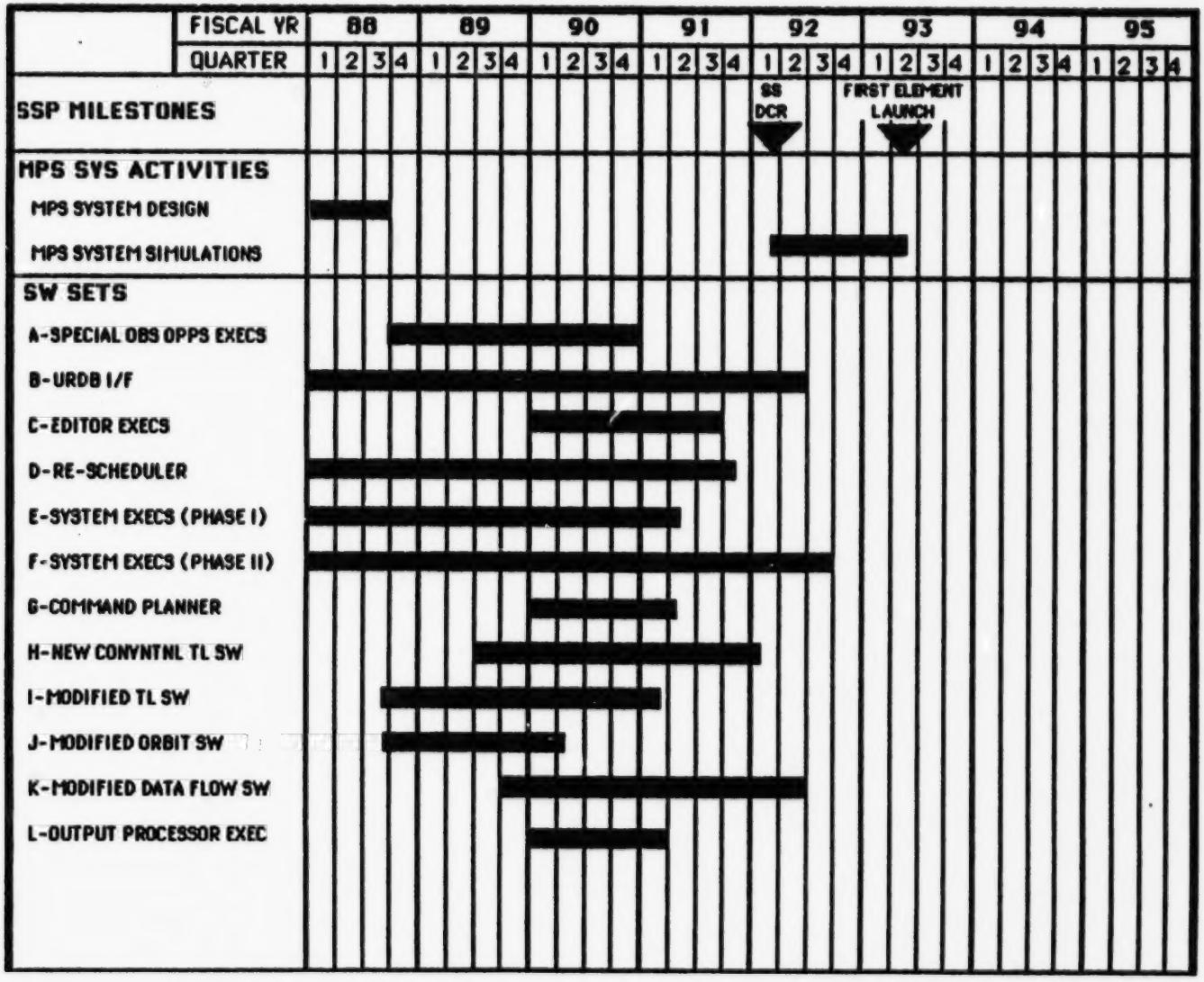


FIGURE 4.3.2-1. SS MPS TOP LEVEL DEVELOPMENT SCHEDULE

SW SETS	CRITICALITY TO SYS PERFORMANCE	SCHEDULE RISK	DEVELOPMENT DURATION(MOS)	OPS CONCEPT DEPENDENCY
A-SPECIAL OBS OPPS EXECS	HIGH	MODERATE	28	LOW
B-URDB I/F	HIGH	HIGH	53	LOW
C-EDITOR EXECS	LOW	LOW	21	MODERATE
D-RE-SCHEDULER	MODERATE	HIGH	46	LOW
E-SYSTEM EXECS (PHASE I)	HIGH	MODERATE	40	MODERATE
F-SYSTEM EXECS (PHASE II)	MODERATE	HIGH	57	MODERATE
G-COMMAND PLANNER	HIGH	LOW	16	HIGH
H-NEW CONVNTNL TL SW	HIGH	LOW	31	HIGH
I-MODIFIED TL SW	HIGH	LOW	30	MODERATE
J-MODIFIED ORBIT SW	HIGH	LOW	20	LOW
K-MODIFIED DATA FLOW SW	HIGH	HIGH	33	HIGH
L-OUTPUT PROCESSOR EXEC	LOW	LOW	15	MODERATE

FIGURE 4.3.2-2. SW SET PHASING CRITERIA

Because of the uncertainties in the SSP overall schedule and the amount of time the SS MPS must be in place before on-orbit payload operations begin, the actual MPS schedule that will be followed is uncertain. The estimates included in this plan are made to scope the magnitude of the project and to provide input for budgetary decisions. As the SS MPS project progresses periodic updates to this schedule will be required.

TABLE 4.3.2-1 MANPOWER LOADING ESTIMATES

FISCAL YR-QUARTER	MANPOWER REQUIRED
1988-1	22
2	22
3	22
4	23
1989-1	28
2	42
3	56
4	73

Section 5

SOFTWARE DEVELOPMENT PROCEDURES

5.1 SW TECHNIQUES AND METHODOLOGIES

The following paragraphs describe the software development techniques and methodologies recommended for use on the SS MPS project.

5.1.1 Structured Analysis for Software Requirements

Structured Analysis (SA) is a method of modeling a system using the DeMarco/Yourdon method of data flow diagramming, process/function descriptions, and data dictionary definitions. This method views a system by the data flows and interfaces and the functions performed upon the data.

This method is a top-down approach where each process/function can be further defined into subprocesses/subfunctions. Also, the data flows (interfaces) can be further divided into subflows. All information is maintained in the data dictionary and process/function descriptions (minispecs) for incorporation into the Software Requirements Specification.

5.1.2 Structured Design for Software Top Level Design

Structured Design (SD), developed by Yourdon and Constantine, is a method of defining the architectural structure of a software system using a top-down method of design. Each module of a system is defined; its interfaces, calling sequence, and location in the hierarchy are all evident using this method. SD also provides sophisticated rules which can be implemented using this method, allowing a designer to;

(1) Minimize data coupling (minimize interfaces)
(2) Maximize module cohesion (keep the functions isolated)
(3) Minimize duplicate code
(4) Separate working modules from the management modules
(5) Simplify implementation
(6) Balance the system

5.1.3 PDL for Software Detailed Design

Program Design Language (PDL) will be the method used for detailed design. PDL is a software tool (see paragraph 5.6) providing a structured English-type language for describing the content of a module or unit. The PDL follows certain rules for describing implementation structure, formatting, and interface descriptions.

5.1.4 Standard for Code Development

Code will be developed in a top-down manner consistent with the standards provided in section 5.5.

5.1.5 Standard for Unit Testing

Unit testing will conform to the informal test plans, where each unit will be tested individually or within groups to demonstrate the implementation meets both the design and requirements.

5.1.6 Standard for Module and Computer Program Integration and Testing

All testing will be done in a top-down manner, unless the criticality of a set of units are such that bottom-up or conglomerate testing would be justified. Each unit will be added in a systematic way such that the executive units are tested first, stubbing out the lower levels, until all units are integrated and tested for the full computer program.

5.1.7 Walkthroughs

Well planned internal walkthroughs will be scheduled during requirements analysis, top level design, detailed design, code and unit test. These walkthroughs will consist of Software Quality Assurance (SQA) and a set of software development peers whose expertise will help ensure the correct implementation and evaluate the design and design trade-offs. The results of these walkthroughs will be recorded and become part of the software development files to track the design decisions and implementation directions.

5.2 SOFTWARE DEVELOPMENT LIBRARY

The SDL will be established during the Preliminary Design activity. The library will consist of a collection of library units established and controlled at each of the facilities used during development and test of the software.

Each library unit will contain the tools, documentation, and source and object code associated with the computer program and phase of development.

Control procedures for each unit comprising the SDL will be thorough and consistent. In cases where duplication exists the unit controlling the master will have been identified.

The following activities will ensure a controlled SDL.

ESTABLISHMENT OF THE SDL

(1) The library unit of each facility will be identified, including its location, hardware host, storage media, and administrator.

(2) The planned contents of the library unit will be identified, including software tools, documentation, and source and object code. At this point, where duplication will exist, the controlling library unit will be identified.

(3) Current contents of each library unit will be inventoried. A controlled list of current contents will be maintained for each library unit.

(4) The naming/numbering schemes for directories, files, data bases, procedures, etc., used by each library unit will be established and published.

(5) The backup procedures and responsibilities will be identified.

(6) All media associated with the library unit will be identified and labeled consistent with the release documentation to which it corresponds.

CONTROL OF THE SDL

(1) User accounts will be established according to library access authorization. Users who have accounts on the system, but are not authorized to modify or access the library elements, will not have privileges associated with their account that would allow them to do so.

(2) Unauthorized use of accounts that have privileges to access and/or modify the library elements will be precluded by the use of passwords.

(3) Where limited access to library elements is authorized, access control lists will be maintained for the files so that groups of users will have varying privileges. For example, access control lists may allow all users to read the file, but only the person responsible for the library unit to write to the file.

5.3 SOFTWARE DEVELOPMENT FILES

Software Development files (SDFs) will be maintained for all computer programs.

5.3.1 Benefits of the Use of SDFs

SDFs provide a means for maintaining software in a manner that is visible, auditable, and consistent across the software development effort. A focal point is established for all information relating to the design, implementation, and unit test of all software elements. The SDFs form the primary review items during walkthroughs and other internal review procedures and the primary management tool for monitoring progress during software development. Documentation included in the SDFs is current and available for incorporation into the deliverable data items.

5.3.2 Responsibility for Maintaining SDFs

Every unit comprising the SS MPS will have a corresponding SDF, though not necessarily in a one-to-one correspondence. One SDF may serve a logically related group of units. he SDF will be created and maintained to a current status by the programmer responsible for the software served by the SDF.

5.3.3 Creation and Maintenance of SDFs

SDFs are created during the Detailed Design phase of software development, after the top level modules have been decomposed into lower level units.

The SDFs will be reviewed by functional and project management and by Software Quality Assurance personnel. Following each event that causes a change to information in the SDF, such as a walkthrough or a unit test, the responsible programmer will update the SDF in a timely manner.

The SDF is a working document (preferably computerized) used during development and test but not maintained after completion of integration and test. SDFs will be retained by the functional and software development organization for historical information.

5.3.4 Contents and Format of the SDFs

The SDF for each unit or logically related group of units will be maintained with the sections described below. Since some of the data that goes into the SDF, such as data flow diagrams and current source code, will be maintained in the Software Development Library, that data may be referenced in the SDF rather than duplicated. However, prior to a review or audit, the responsible programmer will obtain a hard copy of the current version of all referenced data to facilitate the review or audit.

The SDFs will address the following:

(a) schedule
(b) status information
(c) unit requirements
(d) design considerations and constraints
(e) code listing
(f) test documentation

These are addressed in the format below.

COVER SHEETS SECTION

Progress Table

This table indicates the schedule and status information relative to the schedule with sign-off blocks for each milestone enumerated.

Review Items List

The list identifies deficiencies found in the developing software and the engineering responses to those items listed.

Change Log

This log reports all the changes made to the software after the internal baseline has been established.

Walkthrough Reports

This report documents the findings during each walkthrough.

REQUIREMENTS SECTION

This section contains unit requirements including interface requirements that are allocated to the software served by the SDF. Requirements changes which evolve during the software development process will be documented in this section.

DESIGN DESCRIPTION SECTION

This section includes an overview of function detailed design, data base design and interfaces. Any design considerations and constraints will be noted. This section will contain design representation of the software in the form of Program Design Language (PDL) and/or structure charts.

CODE LISTING SECTION

This section will contain a current listing of the source code. It will also contain a reference (filename, directory, and version number) to the last reviewed source code.

UNIT TEST SECTION

This section includes the test methods, cases, tools, and startup conditions for informal testing of the software served by the SDF. Where the SDF corresponds not to a unit but a logical group of units, the test procedures identified here will pertain to the group of units. See paragraph 5.4 for the format of unit test cases and procedures documentation.

UNIT TEST RESULTS SECTION

This section includes the test results and verification that the testing was successfully completed. See paragraph 5.4 for the format of unit test results documentation.

PROBLEM REPORTS SECTION

This section contains copies of all Software Discrepancy Reports (SDRs) generated after the software served by the SDF has been baselined internally.

NOTES SECTION

This section contains any other information, such as memos or records of discussion, that may provide useful information about the software.

5.4 DOCUMENTATION FORMATS FOR INFORMAL TESTS

Informal tests are performed during the Coding and Unit Testing and Module Integration and Testing phases of the software development cycle. Formal tests are tests performed during the Computer Program Testing phase of

the software development cycle. Formats for the documentation of formal tests are recommended to be defined by appropriate Space Station program data requirements for the Software Test Plan (STP), Software Test Description (STD), Software Test Procedure (STPR), and Software Test Report (STR).

Recommended formats for informal test documentation are described in paragraphs 5.4.1 through 5.4.6.

Unit test documentation will be kept in the SDF that serves the unit. Module Integration and Test documentation will be kept, along with the Unit Test documentation, in the SDF that serves that unit belonging to the module which has, among all units for the module, the lowest numbered identifier.

5.4.1 Unit Test Plan/Description

SECTION	CONTENTS
Identification	Identifies the software being tested.
Reference	Identifies the documents essential to an understanding of the test effort.
Schedules and Responsibilities	Identifies those responsible for conducting the tests and the schedules to be followed.
Test Cases	Identifies test cases 1 through X and provides, for each, a summary, objectives, requirements to be verified by the test case, test methods, and acceptance criteria.
Support Requirements	Identifies test tools and test drivers required.

5.4.2 Unit Test Procedures

SECTION	CONTENTS
Identification	Identifies the software being tested.
Objective	Identifies the objective of the test case.
Test Method	Identifies the test environment and step-by-step test procedure.
Inputs	Identifies the test inputs.
Outputs	Identifies the expected outputs.
Support	Identifies any required test tools or test drivers for the test case.

NOTE: Test procedures will be required for each of the test cases identified in the test cases documentation.

5.4.3 Unit Test Report

SECTION	CONTENTS
Identification	Identifies the software being tested.
Summary	Provides a summary of the test results for the test case.
Test Results	Provides output with annotations to indicate where the objectives and requirements are met.
Problems	Identifies and provides an analysis of any deviations from the expected output. Also included are recommendations for corrective action and retest.

NOTE: Test case results will be required for each of the test cases identified in the test cases documentation.

5.4.4 Module Integration Test Cases

The format and required content will be the same as for unit test cases. See paragraph 5.4.1 above.

5.4.5 Module Integration/Test Procedures

The format and required content will be the same as for unit procedures. See paragraph 5.4.2 above.

5.4.6 Module Integration Test Results

The format and required content will be the same as for unit test results. See paragraph 5.4.3 above.

5.5 DESIGN AND CODING STANDARDS

All computer programs will be designed using the top-down approach of the Yourdon Structured Design Methodology, and will be coded in a top-down manner using a Higher Order language (HOL). In general, all new software will be coded in ADA consistent with Space Station program requirements.

Design, coding, and commenting standards are listed below. In cases where a particular standard cannot be implemented in the particular HOL, a convention similar in intent will be used. The language peculiar standards to be applied to the coding of individual computer programs will be stated in the SRS for the computer programs.

Design Standards

Structure Charts shall be used to show the hierarchical structure of the design.

Coding Standards

(1) Each unit shall perform a single function. As a goal, the average number of executable lines per unit shall not exceed 50, with no single unit having more than 100 executable lines of code.

(2) All units shall be comprised of a prologue, declarative statements, and executable statements with comments, in that order.

(3) Modification of a unit's code during execution shall not be permitted.

(4) Except for error exits, each routine shall have a single entry point and a single exit point.

(5) Constants will not be hard-coded in the body of the code but will be defined and assigned a value in the declarations section.

(6) Meaningful names shall be used for constants, variables, functions, and other program elements so that source code listings are readable.

(7) Each line of source code shall contain, at most, one executable statement.

(8) Nesting beyond five levels shall be avoided.

(9) Mixed mode operations (Mixing Integers and Floating Point) shall be avoided; however, if it is necessary to use them, their use shall be documented by conspicuous comments in the source code.

(10) Error and diagnostic messages will be presented in a uniform manner throughout the units and be sufficiently self-explanatory that table hook-ups are not required in order to interpret the message.

(11) Coding conventions will be consistent among all units.

Commenting Standards

(1) Paragraphing, blocking by blank lines, and indenting shall be used to enhance the readability of the code.

(2) Comments shall be set off from the executable source code in a uniform manner.

(3) The following details shall be commented in the prologue section of each program/module/unit:

- Program/module/unit creation date
- Date of last revision, revision number, problem report title and number associated with the change
- Programmer name and department responsible for the unit
- Unit's purpose and how it works

- The PDL generated during detailed design for the unit in comment form
- Functions, performance requirements and external interfaces for the computer program the unit helps implement
- Other units called and the calling sequence
- Data files used including the file name, usage (input, output, or input/output) and summary of use
- Use of global variables, local variables, registers and memory locations
- Any special processing or tasks
- Error conditions for which the code provides

(4) The beginning of executable code shall be indicated by a comment such as "START OF PROGRAM EXECUTION".

(5) Comments will be given in the body of each routine to document each subroutine call, input/output instruction and functional grouping of code.

5.6 SW DEVELOPMENT TOOLS

It is recommended that, as a minimum, the following types of software development tools be utilized in the performance of the SS MPS project:

(1) Virtual memory operating system
(2) Data Base design aid
(3) Requirements specification language and analyzer
(4) Program design language
(5) Programming support library with CM aids
(6) Text editor and manager

Most of the aforementioned types of tools are available in the ADA Programming Support Environment developed by the Stoneman project for the Department of Defense. In addition, for the expert system prototyping the following tools are recommended:

(1) Expert system development tool
(2) Natural language development tool

It is recommended that an in-depth technology survey be performed prior to the purchase of any off-the-shelf AI tools.

APPENDIX A

SW FUNCTIONAL REQUIREMENTS

PRECEDING PAGE BLANK NOT FILMED

A-3 PRECEDING PAGE BLANK NOT FILMED

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
ASEP	MODIFIED	SEE SL MIPS DB. MODIFICATIONS CONSIST OF THE ADDITION OF A ROUTINE TO HANDLE DRAG ANALYTICALLY AND AN OPTION TO WRITE ASCN NODE, DTLD EPHEM AND GND TRCK DATA ON ONE LDF AND EARTH SHADOW ON AN O/O FILE. SHOULD BE SET UP TO PERIODICALLY ACCESS INPUT DATA AND PRODUCE OUTPUT FILES WITH MINIMUM MISSION PLANNERS INTERACTION. INPUT FILES: SOLAR ACTIVITY PROJECTIONS ACTUAL STATE VECTOR OUTPUT FILES: DTLD ORBIT PARAMS LDF EARTH SHADOW O/O	1 SS PROJECTED ORBIT EPHEMERIS GENERATION
TARGEN	MODIFIED	SEE SL MIPS DB. IN THE SS MISSION PLANNING SYSTEM TARGEN WILL BE CALLED AT VARIOUS TIMES TO APPLY CONSTRAINTS AND PERFORM RQD SET THEORY OPERATIONS. MODIFICATIONS RQD WILL CONSIST PRIMARILY OF INTERFACES WITH THE VARIOUS SPECIAL OBSERVATION OPPORTUNITIES EXECUTIVES. INPUT FILES: ANY LIST DIRECTED FILE ANY O/O FILE OUTPUT FILES: AN O/O FILE	2.1 GENERATE SUN RISE/SET 2.5 MERGE STANDARD ORBIT OBS OPPS 3.1.3 COMPUTE TERMINATOR TARGETS 3.1.4 COMBINE CONSTRS AND APPLY FILE MERGE OPPS 3.2.2 COMBINE CONSTRS TO DEFINE SOLAR OBS OPPS 3.3.4 APPLY CONSTRS TO EARTH SITE OBS OPPS 3.4.5 APPLY CONSTRAINTS AND MERGE OPPS 3.5.4 MERGE CELESTIAL OBS OPPS FILES 3.6 MERGE OBS OPPS FILES
STAR	MODIFIED	SEE SL MIPS DB-ADDITIONS/MODIFICATIONS WILL BE TO READ DTLD ORBIT PARAMS FILE INSTEAD OF AN ASCN NODE FILE. PROGRAM WILL ALSO GENERATE SEPARATE NIGHT AND NON-NIGHT CELESTIAL OBS DEFN AND OBS OPPS FILES. CAPABILITY MUST BE PROVIDED TO ENTER DEFNS AND CONSTRAINTS INTERACTIVELY THROUGH THE CELESTIAL OBS OPPS EXECUTIVE. INPUT FILES: DTLD ORBIT PARAMS LDF OUTPUT FILES: MOON RISE/SET O/O CEL OBS DEFNS NDF CEL OBS AC/LOSS O/O	2.2 GENERATE MOON RISE/SET 3.5.1 DEVELOP CELESTIAL OBS DEFNS 3.5.2 GENERATE STELLAR OBS AC/LOS

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
CAVA	MODIFIED	SEE SL MPS DB-A LARGE PORTION OF SL MPS CAVA IS DEVOTED TO HANDLING MANEUVER AND ATT TLS WHICH WILL NOT BE RQD FOR SS.THE ROUTINES APPLICABLE TO TRAJECTORY DATA,SENSOR DATA, TARGET DATA, SENSOR TARGET VISIBILITY, OCCULTATION AND FILE MGMT CAN BE USED WITH MODIFICATIONS AND A MODIFIED DRIVER. INPUT FILES: DTLD ORBIT PARAMS LDF OUTPUT FILES: TDRS AC/LOS O/O FILE	2.3 GENERATE TDRS COVERAGE
RADI2	MODIFIED	SEE SL MPS DB-MUST READ DTLD ORBIT PARAMS FILE INSTEAD OF DTLD EPHEMERIS. INPUT FILES: DTLD ORBIT PARAMS LDF OUTPUT FILES:RAD ENVIR LDF	2.4 GENERATE RADIATION ENVIR-ONMENT
TANRAY	MODIFIED	SEE SL MPS DB-MUST READ DTLD ORBIT PARAMS FILE INSTEAD OF MANUALLY INPUT STATE VECTOR INPUT FILES: DTLD ORBIT PARAMS LDF OUTPUT FILES: TANRAY EPHEM LDF.	3.1.1 COMPUTE DISTANCE FROM SS TO SUN LINE OF SITE TO EARTH SURFACE. COMPUTE SUN RISE/ SET HISTORY.
LTO	MODIFIED	SEE SL MPS DB. IN THE SS MISSION PLANNING SYSTEM LTO WILL BE CALLED AT VARIOUS TIMES TO APPLY ACCEPTANCE CONDITIONS TO A LDF AND PRODUCE AN O/O FILE. MODIFICATIONS RQD WILL CONSIST PRIMARILY OF INTERFACES WITH THE VARIOUS SPECIAL OBSERVATION OPPORTUNITIES EXECUTIVES. INPUT FILES: ANY LIST DIRECTED FILE OUTPUT FILES: AN O/O FILE	3.1.2 DEVELOP/APPLY CONSTRS TO ATMOS PHYS OBS PERIODS 3.2.1 DEVELOP SUN ELEV CONSTRS FOR SOLAR OBS PERIODS 3.3.3 DEVELOP/APPLY EARTH OBS OPPS CONSTRS 3.4.3 DEVLP/APPLY CONSTRS TO BORB PARAMS 3.4.4 GENERATE HEMISPHERE OPPS 3.5.3 IMPOSE RADIATION CONSTRS
ATMOS	MODIFIED	SEE SL MPS DB-WILL READ DTLD ORBIT PARAMS FILE INSTEAD OF ASCN NODE LDF AND EARTH SHADOW O/O FILES. CAPABILITY TO HANDLE VARIABLE ATTITUDES IS NO LONGER NEEDED.MODIFIC-ATIONS RQD WILL CONSIST PRIMARILY OF INTERFACES WITH THE VARIOUS SPECIAL OBSERVATION OPPORTUNITIES EXECUTIVES. INPUT FILES: DTLD ORBIT PARAMS LDF OUTPUT FILES:SUN AZ/ELEV LDF	3.1.5 COMPUTE SUN AZ/ELEV FM SS WRT SUN RISE/SET EVENTS

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
ATMOS PHYS EXEC	NEW	EXPERT SYSTEM EXECUTIVE THAT AIDS THE USER/MSN PLANNER IN DEFINING ATMOSPHERIC PHYSICS OBSERVATION RQMTS AND CALCULATING OBSERVATION OPPORTUNITIES. SYSTEM MUST PROVIDE A USER FRIENDLY INTERFACE WITH ON-LINE HELP AND EXPLANATION FEATURES. THIS EXECUTIVE SHALL SELECT AND SEQUENCE THE APPLICABLE CALCULATION ROUTINES (TANRAY, LTO, TARGEN, ATMOS). ACTUAL ROUTINE CALLS SHALL BE TRANSPARENT TO THE USER. THE USER SHALL BE PROVIDED WITH A "GENERIC" SET OF INPUT DEFAULT VALUES THAT ARE CONSTANTLY UPDATED BASED ON USER INPUTS TO OTHER FIELDS. THIS PROVIDES A WORKING MODEL OF USER OBS OPPS REQUIREMENTS REGARDLESS OF THE AMOUNT OF USER DEFINITION. THE EXECUTIVE SHALL HAVE THE CAPABILITY TO RECOGNIZE REQUESTS OUTSIDE ITS CURRENT KNOWLEDGE DOMAIN AND REQUEST ASSISTANCE FROM THE USER/OPERATOR.	3.1.6 EXEC FOR ATMOS PHYS OBS OPPS
ESDAT	MODIFIED	SEE SL MIPS DB-GND SITE DEFNS WILL BE INPUT THROUGH EARTH SITE EXECUTIVE. INPUT FILES: NONE OUTPUT FILES: SITE AC/LOS O/O	3.3.1 CREATE EARTH SITE DEFN FILE
ESAL	MODIFIED	SEE SL MIPS DB-WILL READ DTLD ORBIT PARAMS LDF INSTEAD OF ASCN NODE. INPUT FILES: SITE DEFNS NDF OUTPUT FILES: SITE AC/LOS O/O	3.3.2 GENERATE AREA SITE AC/LOS

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
EARTH SITE EXEC	NEW	EXPERT SYSTEM EXECUTIVE THAT AIDS THE USER/MSN PLANNER IN DEFINING EARTH SITE OBSERVATION RQMTS AND CALCULATING OBSERVATION OPPORTUNITIES. SYSTEM MUST PROVIDE A USER FRIENDLY INTERFACE WITH ON-LINE HELP AND ALSO INTERFACE WITH THE APPLICABLE CALCULATION ROUTINES (ESDAT, ESAL, LTO,TARGEN,TAE). ACTUAL ROUTINE CALLS SHALL BE AS TRANSPARENT AS POSSIBLE TO THE USER. THIS EXEC CONTAINS FEATURES IDENTICAL TO THE ATMOS PHYS. EXECECUTIVE.	3.3.6 EXEC FOR EARTH SITE OBS OPPS
TAE	MODIFIED	SEE SL MIPS DB- STATISTICAL ANALYSIS ROUTINE WILL BE CALLED BY ORBITAL ANALYSIS EXECUTIVE PROGRAMS. O/O FILES WILL NOT BE REFORMATED AND BUILDING NEW SUBJECTS AND EDITING DATA WILL BE DONE IN THE O/O FILE FORMAT. MUST ALSO BE MODIFIED TO EXTRACT USER TIME PREFERENCES DIRECTLY FROM THE URDB. INPUT FILES: MSN OBS OPPS O/O OR URDB OUTPUT FILES: MSN OBS OPPS O/O	3.3.5 STATISTICAL ANAL OF OBS OPPS 3.4.6 STATISTICAL ANAL OF OBS OPPS 3.5.5 STATISTICAL ANAL OF OBS OPPS 5.1.1 EXTRACT USR TIME PREFERENCES 5.1.2 BUILD NEW SUBJECTS IF RQD 5.1.3 STATISTICAL ANALYSIS OF OBS OPPS 5.1.4 OBS OPPS DATA EDITING 8.1.1 EXTRACT USR TIME PREFERENCES 8.1.2 BUILD NEW SUBJECTS IF RQD 8.1.3 STATISTICAL ANALYSIS OF OBS OPPS 8.1.4 OBS OPPS DATA EDITING
BORB	MODIFIED	SEE SL MIPS DB-MODIFIED TO READ DTLD ORBIT PARAMS LDF INSTEAD OF ASCN NODE FILE. CAPABILITY TO HANDLE ATT TL O/O FILE IS NO LONGER REQUIRED. PROGRAM WILL BE DRIVEN BY PLAMSA PHYS EXECUTIVE. INPUT FILES:DTLD ORBIT PARAMS LDF OUTPUT FILES: BORB PARAMS LDF	3.4.1 COMPUTE ORIENTATION AND STRENGTH OF MAGNETIC FIELD IN SS BODY COORD SYS 3.4.2 DEVELOP PLASMA PHYSICS OBS DFNS

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
PLASMA PHYS EXEC	NEW	EXPERT SYSTEM EXECUTIVE THAT AIDS THE USER/MSN PLANNER IN DEFINING PLASMA PHYS OBSERVATION RQMTS AND CALCULATING OBSERVATION OPPORTUNITIES. SYSTEM MUST PROVIDE A USER FRIENDLY INTERFACE WITH ON-LINE HELP AND ALSO INTERFACE WITH THE APPLICABLE CALCULATION ROUTINES . (BORB, LTO, TARGEN,TAE). ACTUAL ROUTINE CALLS SHALL BE AS TRANSPARENT AS POSSIBLE TO THE USER. THIS EXECUTIVE HAS FEATURES IDENTICAL TO THE ATMOS PHYSICS EXECUTIVE.	3.4.2 DEVELOP PLASMA PHYSICS OBS DEFNS 3.4.7 EXEC FOR PLASMA PHYS OBS OPPS
CELEST-IAL EXEC	NEW	EXPERT SYSTEM EXECUTIVE THAT AIDS THE USER/MSN PLANNER IN DEFINING CELESTIAL OBSERVATION RQMTS AND CALCULATING OBSERVATION OPPORTUNITIES. SYSTEM MUST PROVIDE A USER FRIENDLY INTERFACE WITH ON-LINE HELP AND ALSO INTERFACE WITH THE APPLICABLE CALCULATION ROUTINES (STAR, TARGEN, LTO, TAE). ACTUAL ROUTINE CALLS SHALL BE AS TRANSPARENT AS POSSIBLE TO THE USER. THIS EXECUTIVE HAS FEATURES IDENTICAL TO THE ATMOS PHYSICS EXECUTIVE.	3.5.6 EXEC FOR STELLAR OBS OPPS
VME	MODIFIED	SEE SL MIPS DB-INTERFACE MUST BE MODIFIED FOR COMPATIBILITIY WITH MODEL EDITOR EXECUTIVE. INPUT FILES: USER MDLS SS OPS CONSTRS DB OUTPUT FILES: ESS MODELS	5.2.3 EDIT USER MSN TL MODELS 5.2.4 BUILD CREW CYCLE SYSTEM MODELS 8.2.3 EDIT USER MSN TL MODELS 8.2.4 BUILD CREW CYCLE SYSTEM MODELS 7.3 ASSIGN USER RSC ENV ALLOCS

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
OBS OPPS EDITOR EXEC	NEW	EXPERT SYSTEM EXECUTIVE THAT GUIDES THE USER/MISSION PLANNER IN BUILDING SPECIAL OBSERVATION OPPORTUNITIES. ALLOWS USER TO INPUT REQUIRED OPERATING TIMES THAT ARE A RESULT OF SOMETHING OTHER THAN AN ORBITAL OPPORTUNITY. MUST INTERFACE WITH TAE ROUTINE THAT BUILDS NEW SUBJECTS. THE EXECUTIVE MUST HAVE THE ABILITY TO CONSTRUCT A VALID OBS OPPS SUBJECT IF REQUIRED, AND EXPLAIN THE PROCESS TO THE MISSION PLANNER. IT SHOULD BE CAPABLE OF ACTIVATING THE STATISTICAL ANALYSIS ROUTINE AND ASSESSING THE RESULTS FOR CONFIDENCE AND RELIABILITY OF THE OUTPUT WINDOW OF OBS OPPS.	5.1.5 OBS OPPS EDITOR EXEC 8.1.5 OBS OPPS EDITOR EXEC
MODEL EDITOR EXEC	NEW	EXPERT SYSTEM TO AID MISSION PLANNERS IN DEVELOPING MISSION TIMELINE MODELS. MUST PROVIDE INTERFACE WITH VME. THE EXECUTIVE SHOULD PERFORM ALL SOFTWARE MODULE SELECTION AND SEQUENCING AND PROVIDE GUIDELINES AND DEFAULT EXAMPLES FOR CONSTRUCTING TIMELINE MODELS. IT SHOULD MONITOR ALL TL MODELS FOR INTERNAL CONSISTENCY AND CONSTRAINTS. AN EXPLANATORY, USER FRIENDLY INTERFACE SHOULD BE PROVIDED.	5.2.5 MODEL EDITOR EXECUTIVE 8.2.5 MODEL EDITOR EXECUTIVE
MDL EXTRACT	NEW	INTERFACE WITH THE USER REQUIREMENTS DATA BASE AND EXTRACT THE APPROPRIATE USER TL MODELS DATA. INPUT FILES: URDB OUTPUT FILES: USER MODELS	5.2.1 EXTRACT USER TL MDLS FROM DB 8.2.1 EXTRACT USER TL MDLS FROM DB
MDL COMPARE	NEW	VERIFY COMPATIBILITY OF USER MODELS WITH USER RESOURCE ALLOCATIONS. INPUT FILES: USER MODELS USER RSC ENV ALLOCS OUTPUT FILES: USER MODELS	5.2.2 VERIFY COMPATIBILITY 8.2.2 VERIFY COMPATIBILITY

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
SCHED-ULER EXEC	NEW	EXPERT SYSTEM EXECUTIVE THAT WILL GUIDE MISSION PLANNERS AT BOTH THE PLANNING CENTERS AND PAYLOAD INTEGRATION CENTER IN USE OF ALL SCHEDULING SW MODULES. MAJOR MODULES WILL CONSIST OF A REVISED VERSION OF ESP, A NEW EXPERT RE-SCHEDULER AND VARIOUS UTILITY ROUTINES TO ALLOW COMBINING SCHEDULES, COMPARING SCHEDULES, AND GENERATING INTERFACE FILES ETC. MODULES AND UTILITY ROUTINES CURRENTLY IDENTIFIED ARE DISCUSSED INDIVIDUALLY. THE EXECUTIVE SHOULD MONITOR ALL I/O FOR CONSISTENCY AND INTERMODULE CONSTRAINTS. IT SHOULD BE CAPABLE OF PERFORMING CONFIDENCE ASSESSMENTS (E.G. 95%) OF ALL THE OUTPUT REQUIREMENTS AND IDENTIFY INPUTS THAT ARE NOT STRONGLY SUPPORTED BY THE CURRENT KNOWLEDGE BASE.	5.3.5 SCHEDULER EXECUTIVE 8.3.5 SCHEDULER EXECUTIVE
ESP	MODIFIED	SEE SL MIPS DB THE BASIC SCHEDULING PROCESS IS THE SAME AS SL MIPS ESP. THE CORE OF THE SCHEDULER SHOULD REQUIRE ONLY MINOR MODIFICATIONS. ADDITIONS/MODIFICATIONS WILL EXIST PRIMARILY IN THE AREAS OF INTERFACE FILES WITH THE ES RE-SCHEDULER AND THE OVERALL SCHEDULER EXECUTIVE. INPUT FILES: MSN OBS OPPS O/O ESS MODELS RESOURCE ALLOCS FILE(PLNG CTR BASIS OR PLD INTEG CTR BASIS) OUTPUT FILES: PLNG CTR GROSS USER OPS PLNG CTR CREW ACT PLANS INHB/CONSTR RQMTS SCHDLD USER RSC ALLOCS INTEG PLD CNSLDTD SCHDL	5.3.1 SCHED CREW CYCLE, SYSTEM MODELS 5.3.2 SCHED USER OPERATIONS 5.3.4 GENERATE OUTPUT 6.2 VFY COMPATIBILITY, IDENTIFY DEVIATIONS 6.4 EXTRACT RESOURCE REQUIREMENTS 8.3.1 SCHED CREW CYCLE, SYSTEM MODELS 8.3.2 SCHED USER OPERATIONS 8.3.4 GENERATE OUTPUT 9.3 VERIFY COMPATIBILITY WITH SS OPS 9.4 IDENTIFY CONFLICTS ACROSS PLNG CTRS 9.6 GENERATE INTGD PLD TOP 10 GENERATE USER ACTIVITY PLANS 11.3 VERIFY RSC AND OPS CONSTRS CONFLICTS 11.5 GEN PLNG CTR ACT PLAN 12.3 PERFORM INTEGRATED RSC CHECK/IDENTIFY CONFLICTS 12.5 GENERATE INTEG'D PYLD ACT PLAN

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
TL COMPARE	NEW	UTILITY ROUTINE THAT WILL COMPARE--AND IDENTIFY DEVIATIONS--A RESOURCE ALLOCATIONS PROFILE WITH A SCHEDULE THAT WAS DEVELOPED TO FIT WITHIN THOSE ALLOCATIONS FOR VERIFICATION PURPOSES. MUST INTERFACE UPWARDS WITH THE SCHEDULER EXECUTIVE. INPUT FILES: RESOURCE ALLOCATION FILES ESS TL FILES(USERS OR PLNG CTRS) OUTPUT FILES: VERIFIED ESS TL FILES WITH DEVIATIONS	7.1 COMPARE, IDENTIFY DEVIATIONS 9.1 VERIFY COMPATIBILITY, IDENTIFY DEVIATIONS 11.1 VERIFY COMPATIBILITY,IDENTIFY DEVIATIONS 12.1 VERIFY COMPATIBILITY, IDENTIFY DEVIATIONS
TL MERGE	NEW	UTILITY ROUTINE THAT WILL CONSOLIDATE VERIFIED SCHEDULES/TIMELINES INTO AN INTEGRATED SCHEDULE OF THE ESS FORMAT. MUST INTERFACE UPWARDS WITH THE SCHEDULER EXECUTIVE. INPUT FILES: ESS TL FILES OUTPUT FILES: CONSOLIDATED ESS TL FILES	6.1 CONSOLIDATE REQUIREMENTS 9.2 CONSOLIDATE PLANNING CTR TOPS 11.2 CONSOLIDATE USER ACT PLANS 12.2 CONSOLIDATE PLNG CTR ACT PLANS
CMD PLANNER	NEW	USER FRIENDLY EDITOR TO GUIDE/ASSIST USERS IN BUILDING USER SPECIFIC COMMAND LISTS COMPLETE WITH DELTA TIMES. ALSO TAKES ACTVITY PLANS AS INPUT AND TIMETAGS COMMAND LISTS WITH ABSOLUTE TIMES. THE EXECUTIVE SHOULD RELY ON INPUT FROM THE URDB TO ASSEMBLE RECOMMENDED COMMAND SEQUENCES BASED ON CONSTRAINTS AND THE CURRENT EXPERIENCE BASE. IT SHOULD ALSO BE ABLE TO PERFORM REORGANIZATION OF GROUPS OF COMMANDS BASED ON USER INPUTS. INPUT FILES: URDB USER ACTIVITY PLANS DB OUTPUT FILES: USER CMD PLANS DB	13.1 CREATE COMMAND LIST 13.2 PRODUCE TIMETAGS 13.3 GENERATE CMD TL OUTPUT

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
RE-SCHED-ULER	NEW	THIS MODULE IS AN EXPERT SYSTEM THAT WILL TAKE A SCHEDULE/TIMELINE GENERATED BY ESP AND ALLOW RE-SCHEDULING OF OPERATIONS TO REFINE THE ORIGINAL SCHEDULE, MAKE CHANGES REQUIRED AS A RESULT OF OVERALL PAYLOAD INTEGRATION OR ALLOW ON-BOARD RE-SCHEDULING BY THE CREW. THIS MODULE WILL READ AN ESS FORMATED TIMELINE, AID THE OPERATOR IN RE-SCHEDULING OPERATIONS AND CREATE AN OUTPUT FILE IN THE ESS FORMAT. PLANNET AND MAESTRO ARE POTENTIAL BASELINE MODELS FOR THIS MODULE. SINCE PLANS ARE FOR ONE VERSION OF THIS MODULE TO RESIDE ON-BOARD THE SS RQMT THAT ALL FLIGHT SW BE WRITTEN IN ADA MUST BE CONSIDERED. INPUT FILES: ESS TL FILES ESS MODEL FILES OUTPUT FILES: UPDT'D ESS TL FILES	5.3.3 RE-SCHEDULE USER OPS(ES) 8.3.3 RE-SCHEDULE USER OPS(ES) 9.5 RE-SCHEDULE USER OPS TO ELIMINATE CONFLICTS BETWEEEN PLNG CTRS 11.4 RE-SCHEDULE OPS TO ELIMINATE CONFLICTS BETWEEN USERS 12.4 RE-SCHEDULE OPS TO ELIMINATE CONFLICTS BETWEEN PLNG CTRS 15.1 ON-BOARD RE-SCHEDULING
PCAP	MODIFIED	SEE SL MPS DB- SS OPERATIONS MAY IMPOSE NEW RQMTS ON THE LAYOUT OF THE PCAP CHARTS. INTERFACE FILES WILL BE SOMEWHAT DIFFERENT AND HARDCOPY OUTPUT WILL BE OPTIONAL. INPUT FILES: CREW PRCDRS DB ESS MODELS FILE PLNG CTR USER OPS MSN OBS OPPS O/O OUTPUT FILES: PCAP CHARTS FILE NOTES FILE PROCEDURES FILE	11.6.1 GENERATE PCAP CHARTS
PTS	MODIFIED	SEE SL MPS DB- SS OPERATIONS MAY IMPOSE NEW RQMTS ON THE LAYOUT OF THE PTS CHARTS. INTERFACE FILES WILL BE SOME-WHAT DIFFERENT AND HARDCOPY OUTPUT WILL BE OPTIONAL. INPUT FILES: PLNG CTR CREW ACT PLANS DTLD ORBIT PARAMS LDF INHB/CONSTR RQMTS PLNG CTR USER OPS MSN OBS OPPS O/O OUTPUT FILES: PTS CHARTS FILE	11.6.2 GENERATE PTS CHARTS

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
SUMMARY PCAP	NEW	PROGRAM THAT WILL GENERATE A SUMMARY PCAP FROM THE CONSOLIDATED PAYLOAD OPERATIONS SCHEDULE AND THE DETAILED PLANNING CENTER DAILY PCAPS. INPUT FILES:DTLD PLNG CTR DAILY PCAP CNSLDTD PYLD ESS TL FILE OUTPUT FILES: PYLD SUMRY PCAP	12.6 GENERATE INTEGRATED PAYLOAD SUMMARY PCAP
PCAP DELTAS	NEW	ON-BOARD SW THAT WILL, AFTER CREW CHANGES ARE MADE TO THE INDIVIDUAL PLANNING CENTER TIMELINES, MODIFY THE EXISTING ON-BOARD PCAP, SEND THE UPDATED PCAP TO THE ON-BOARD SS ACTIVITY PLANS DATA BASE AND DOWNLINK A DELTAS FILE THAT WILL ALLOW GROUND PERSONNEL TO UPDATE THEIR VERSION OF THE PCAP. INPUT FILES: PLNG CTR DTLD DAILY PCAP ESS TL FILE OUTPUT FILES: UPDTD PLNG CTR DTLD PCAP PLNG CTR DTLD PCAP DELTAS PYLD SUMRY PCAP	15.2 UPDATE PLNG CTR DTLD PCAP
PLNG CTR MISSION PLNG EXEC	NEW	PHASE I: A HIGH LEVEL EXECUTIVE WHICH PROVIDES MISSION PLANNING PERSONNEL A STANDARDIZED SYSTEM FOR USE OF LOWER LEVEL EXECUTIVES AND CALCULATION ROUTINES AS WELL AS PROVIDING A FILE MANAGEMENT SYSTEM. THE NATURAL LANGUAGE INTERFACE DESCRIBED IN THE USER PLANNING EXECUTIVE WILL BE AVAILABLE WITH A VOCABULARY TAILORED TO THE PLANNING CENTER APPLICATIONS. PHASE II: CAPABILITY WILL BE PROVIDED SUCH THAT LOGIC MAY BE ENCODED IN SUBFUNCTIONS 5, 7, 8 AND 10 OF PLANNING CYCLES A, B, AND C AND USED AS AN ADVISOR IN SUBFUNCTIONS 8 AND 10 OF THE REPLANNING CYCLE D.	

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
PYLD OPS INTEG. CTR MSN PLNG EXEC	NEW	PHASE I: A HIGH LEVEL EXECUTIVE WHICH PROVIDES MISSION PLANNING PERSONNEL A STANDARDIZED SYSTEM FOR USE OF LOWER LEVEL EXECUTIVES AND CALCULATION ROUTINES AS WELL AS PROVIDING A FILE MANAGEMENT SYSTEM. SIMILAR TO THE PLNG CTR VERSION. PHASE II: CAPABILITY WILL BE PROVIDED SO THAT LOGIC MAY BE ENCODED DURING SUBFUNCTIONS 6, 9, 11, AND 12 OF PLANNING CYCLES A, B, C AND USED AS AN ADVISOR IN SUBFUNCTIONS 6, 11, AND 12 OF THE REPLANNING CYLE D.	
SPECIAL OBS OPPS EXEC	NEW	A HIGH LEVEL EXECUTIVE THAT INTERFACES UPWARDS TO THE USER AND PLANNING CENTER MISSION PLANNING EXECUTIVES AND DOWNWARD TO THE INDIVIDUAL DISCIPLINE OBSERVATION OPPORTUNITIES EXECUTIVES. THE BASIC FUNCTION IS TO IDENTIFY THE APPLICABLE DISCIPLINE(S) IMPLIED BY THE USER INPUT OBS OPPS DEFINITIONS. THE APPLICABLE DISCIPLINE EXECUTIVES ARE ACTIVATED TO PROVIDE THE DETAILED OBS OPPS DEFINITION. OTHER MODULES ARE POLLED TO IDENTIFY ANY POSSIBLE CONSTRAINTS. THE EXECUTIVE SHALL PROVIDE A USER FRIENDLY INTERFACE WITH BUILT IN TRAINING AND EXPLANATION FEATURES. IT SHALL BE ABLE TO ORGANIZE AND MANIPULATE THE OBS OPPS SETS FOR BEST FIT EVALUATIONS BY THE USER. IT SHALL BE ABLE TO TAG SELECTED SETS FOR LATER RECALL BY THE MISSION PLANNER.	3.7 TOP LEVEL SPECIAL SPECIAL OBS OPPS EXECUTIVE 3.8 EXTRACT USER OBS OPPS RQMTS DATA

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
USER MSN PLNG EXEC	NEW	PHASE I: A HIGH LEVEL EXECUTIVE THAT PROVIDES A USER FRIENDLY INTERFACE FOR SS USERS TO THE USER MISSION PLANNING SOFTWARE RESIDENT AT THE PLANNING CENTERS. THE EXECUTIVE WILL PROVIDE CURRENT MISSION INCREMENT INFORMATION SUCH AS DATES/TIMES WHEN USER ROMTS/DETAILED SCHEDULES MUST BE COMPLETED, A GENERAL OVERVIEW OF THE MISSION PLANNING PROCESS AND PROVIDE A HIGH LEVEL GUIDE TO THE USE OF THE APPROPRIATE MISSION PLANNING SW MODULES. THE USER SHALL BE ABLE TO DIALOG WITH THE EXECUTIVE VIA A NATURAL LANGUAGE INTERFACE. THE EXECUTIVE MUST BE ABLE TO MAKE ASSUMPTIONS RELIABLY AND PERFORM CONSISTENCY AND CONSTRAINT CHECKING ON ALL USER INPUT/OUTPUT. IT MUST BE CAPABLE OF RECOGNIZING ITS KNOWLEDGE DOMAIN LIMITATIONS AND REQUESTING USER/OPERATOR ASSISTANCE WHEN REQUIRED. PHASE II: ADD THE CAPABILITY FOR THE EXECUTIVE TO EXTRACT THE REASONING BEHIND THE INPUT DECISIONS SPECIFIED BY THE USER. THESE REASONS WILL BE ENCODED AND MODELED FOR USE IN AN ADVISORY CAPACITY DURING THE MISSION REPLANNING CYCLE.	

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
SOLAR EXEC	NEW	EXPERT SYSTEM EXECUTIVE THAT AIDS THE USER/MSN PLANNER IN DEFINING SOLAR OBSERVATION RQMTS AND CALCULATING OBSERVATION OPPORTUNITIES. SYSTEM MUST PROVIDE A USER FRIENDLY INTERFACE WITH ON-LINE HELP AND EXPLANATION FEATURES. THIS EXECUTIVE SHALL SELECT AND SEQUENCE THE APPLICABLE CALCULATION ROUTINES (LTO, TARGEN). ACTUAL ROUTINE CALLS SHALL BE TRANSPARENT TO THE USER. THE USER SHALL BE PROVIDED WITH A "GENERIC" SET OF INPUT DEFAULT VALUES THAT ARE CONSTANTLY UPDATED BASED ON USER INPUT TO OTHER FIELDS. THIS PROVIDES A WORKING MODEL OF USER OBS OPPS REQUIREMENTS REGARDLESS OF THE AMOUNT OF USER DEFINITION. THE EXECUTIVE SHALL HAVE THE CAPABILITY TO RECOGNIZE REQUESTS OUTSIDE ITS CURRENT KNOWLEDGE DOMAIN AND REQUEST ASSISTANCE FROM THE USER/OPERATOR.	3.2.3 EXEC FOR SOLAR OBS OPPS
OUTPUT PROCESS-OR EXEC	NEW	EXECUTIVE PROGRAM THAT AIDS THE MISSION PLANNER IN THE USE OF THE PTS, PCAP, AND SUMMARY PCAP PROGRAMS. MUST INTERFACE UPWARDS WITH THE PLANNING CENTER AND PAYLOAD OPERATIONS INTEGRATION CENTER SYSTEM EXECUTIVES.	

SS MPS SOFTWARE REQUIREMENTS SUMMARY

SW MODULE NAME	NEW OR MODIFIED	SW MODULE FUNCTIONAL DESCRIPTION	APPLICABLE SUBFUNCTION/TASK
URDB I/F	NEW	EXPERT SYSTEM THAT GUIDES/PROMPTS USERS IN ENTERING FUNCTIONAL REQUIREMENTS INTO THE DATA BASE TO PROVIDE MISSION PLANNERS THE APPROPRIATE INFORMATION FOR PLANNING AND SCHEDULING. THE SYSTEM SHOULD ALLOW INTERACTIVE FORM EDITING BY THE USER WITH ON-LINE HELP, DATA ENTRY RULES AND MEANINGFUL DEFAULT VALUES. THE TYPES OF REQUIREMENTS TO BE INCLUDED ARE; RQD RESOURCE VECTORS (POWER, CREW,THERMAL, DATA,ETC.); RQD OBSERVATIONS DEFINITIONS; OPERATIONAL CONSTRAINTS(INHIBITS, ETC.); SEQUENCING,CONCURRENCY RQMTS; AND MIN/MAX #OF PERFORMANCES, DURATIONS. THE DB INTERFACE MUST PROVIDE THE CAPABILITY TO RECOGNIZE REQUESTS/INPUTS OUTSIDE OF ITS KNOWLEDGE DOMAIN AND REQUEST HUMAN EXPERT ASSISTANCE WHEN KNOWLEDGE BASE IS INADEQUATE. THE SYSTEM SHOULD BE ABLE TO INTELLIGENTLY UPGRADE DEFAULT VALUES BASED UPON LATEST INPUT DATA FROM USER. THE SYSTEM SHOULD GENERALIZE LOWER LEVEL DETAILS INTO UPPER LEVEL REQUIREMENTS. ALL REQUIREMENTS SHOULD BE CHECKED FOR CONSTRAINTS IN ALL SIX DISCIPLINES. OUTPUT REQUIREMENTS SHALL HAVE ALL ASSUMPTIONS NOTED AND CONFIDENCE FACTORS (E.G. 90%) ASSIGNED.	4 USER REQUIREMENTS DATA BASE I/F

Appendix B
COCOMO SW Set Cost Estimates

This appendix contains the COCOMO detailed cost estimate for each SW Set. This output was used as the basis for the SS MPS cost estimate.

A- SPECIAL OBSERVATION OPPORTUNITIES EXECUTIVES

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	28.8	5.9	$0	4.9
Product Design	64.7	5.9	$0	10.9
Programming	183.3	5.9	$0	31.0
Integration & Testing	111.4	4.6	$0	24.2
TOTAL	388.2	22.4	$0	17.4

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

	SOFTWARE PHASES				
SOFTWARE ACTIVITIES	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	12.65	6.47	5.50	2.23	26.85
Product Des	4.31	27.18	11.00	4.46	46.95
Programming	2.30	8.41	100.83	49.03	160.58
Test Plng	1.44	4.53	12.83	4.46	23.26
V & V	2.59	5.82	20.17	25.63	54.21
Project Off	2.88	5.82	11.00	7.80	27.50
CM/QA	1.15	1.94	12.83	10.03	25.95
Manuals	1.44	4.53	9.17	7.80	22.93
TOTAL	28.76	64.71	183.34	111.44	388.24

END OF OUTPUT SECTION 3

PRECEDING PAGE BLANK NOT FILMED

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	32.4	6.1	$0	5.3
Product Design	72.9	6.1	$0	11.9
Programming	206.6	6.1	$0	33.6
Integration & Testing	125.6	4.8	$0	26.3
TOTAL	437.5	23.2	$0	18.8

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

	SOFTWARE PHASES				
SOFTWARE ACTIVITIES	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	14.26	7.29	6.20	2.51	30.26
Product Des	4.86	30.63	12.40	5.02	52.91
Programming	2.59	9.48	113.63	55.26	180.96
Test Plng	1.62	5.10	14.46	5.02	26.21
V & V	2.92	6.56	22.73	28.88	61.09
Project Off	3.24	6.56	12.40	8.79	30.99
CM/QA	1.30	2.19	14.46	11.30	29.25
Manuals	1.62	5.10	10.33	8.79	25.85
TOTAL	32.41	72.92	206.61	125.58	437.52

C- EDITOR EXECUTIVES

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	11.2	3.9	$0	2.9
Product Design	25.3	4.1	$0	6.1
Programming	75.8	4.9	$0	15.6
Integration & Testing	39.3	3.2	$0	12.4
TOTAL	151.7	16.1	$0	9.4

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

	SOFTWARE PHASES				
SOFTWARE ACTIVITIES	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	5.17	2.53	2.28	0.79	10.76
Product Des	1.57	10.62	4.55	1.57	18.31
Programming	0.67	3.03	41.71	15.73	61.15
Test Plng	0.45	1.52	4.55	1.57	8.09
V & V	0.90	2.02	7.58	9.83	20.34
Project Off	1.35	2.78	5.31	3.15	12.58
CM/QA	0.45	0.76	5.31	3.54	10.06
Manuals	0.67	2.02	4.55	3.15	10.39
TOTAL	11.24	25.28	75.84	39.33	151.69

END OF OUTPUT SECTION 3

D- RE-SCHEDULER

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	27.7	5.8	$0	4.7
Product Design	62.4	5.8	$0	10.7
Programming	176.8	5.8	$0	30.2
Integration & Testing	107.4	4.5	$0	23.6
TOTAL	374.3	22.1	$0	16.9

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

SOFTWARE ACTIVITIES	SOFTWARE PHASES				
	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	12.20	6.24	5.30	2.15	25.89
Product Des	4.16	26.20	10.61	4.30	45.26
Programming	2.22	8.11	97.21	47.27	154.82
Test Plng	1.39	4.37	12.37	4.30	22.42
V & V	2.50	5.61	19.44	24.71	52.26
Project Off	2.77	5.61	10.61	7.52	26.51
CM/QA	1.11	1.87	12.37	9.67	25.02
Manuals	1.39	4.37	8.84	7.52	22.11
TOTAL	27.73	62.38	176.75	107.44	374.30

E-SYSTEM EXECUTIVES (PHASE I)

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	28.7	5.9	$0	4.8
Product Design	64.5	5.9	$0	10.9
Programming	182.7	5.9	$0	30.9
Integration & Testing	111.0	4.6	$0	24.2
TOTAL	386.8	22.3	$0	17.3

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

	SOFTWARE PHASES				
SOFTWARE ACTIVITIES	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	12.61	6.45	5.48	2.22	26.75
Product Des	4.30	27.08	10.96	4.44	46.78
Programming	2.29	8.38	100.46	48.85	159.99
Test Plng	1.43	4.51	12.79	4.44	23.17
V & V	2.58	5.80	20.09	25.54	54.01
Project Off	2.87	5.80	10.96	7.77	27.40
CM/QA	1.15	1.93	12.79	9.99	25.86
Manuals	1.43	4.51	9.13	7.77	22.85
TOTAL	28.65	64.47	182.66	111.03	386.81

END OF OUTPUT SECTION 3

F- SYSTEM EXECUTIVES (PHASE II)

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAG STAFF
Plans & Requirements	46.1	6.9	$0	[illegible]
Product Design	103.7	6.9	$0	15
Programming	293.9	6.9	$0	4[illegible]
Integration & Testing	178.7	5.4	$0	3[illegible]
TOTAL	622.4	26.0	$0	23[illegible]9

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

SOFTWARE ACTIVITIES	SOFTWARE PHASES				
	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTA[illegible]
Reqts Anlys	20.29	10.37	8.82	3.57	4[illegible]5
Product Des	6.92	43.57	17.64	7.15	7[illegible]7
Programming	3.69	13.49	161.66	78.61	25[illegible]5
Test Plng	2.31	7.26	20.57	7.15	[illegible]29
V & V	4.15	9.34	32.33	41.09	[illegible]1
Project Off	4.61	9.34	17.64	12.51	[illegible]09
CM/QA	1.84	3.11	20.57	16.08	[illegible]61
Manuals	2.31	7.26	14.70	12.51	36.77
TOTAL	46.11	103.74	293.93	178.66	622.43

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	5.0	3.0	$0	1.7
Product Design	11.2	3.2	$0	3.5
Programming	33.7	3.8	$0	9.0
Integration & Testing	17.5	2.4	$0	7.2
TOTAL	67.3	12.4	$0	5.4

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

SOFTWARE ACTIVITIES	SOFTWARE PHASES				
	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	2.29	1.12	1.01	0.35	4.78
Product Des	0.70	4.71	2.02	0.70	8.13
Programming	0.30	1.35	18.51	6.98	27.14
Test Plng	0.20	0.67	2.02	0.70	3.59
V & V	0.40	0.90	3.37	4.36	9.03
Project Off	0.60	1.23	2.36	1.40	5.59
CM/QA	0.20	0.34	2.36	1.57	4.46
Manuals	0.30	0.90	2.02	1.40	4.61
TOTAL	4.99	11.22	33.66	17.45	67.32

END OF OUTPUT SECTION 3

H-NEW CONVENTIONAL TIMELINE SOFTWARE

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	36.9	6.4	$0	5.8
Product Design	83.0	6.4	$0	13.0
Programming	235.2	6.4	$0	36.7
Integration & Testing	143.0	5.0	$0	28.7
TOTAL	498.1	24.2	$0	20.6

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

	SOFTWARE PHASES				
SOFTWARE ACTIVITIES	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	16.23	8.30	7.06	2.86	34.45
Product Des	5.53	34.87	14.11	5.72	60.23
Programming	2.95	10.79	129.36	62.91	206.01
Test Plng	1.84	5.81	16.46	5.72	29.84
V & V	3.32	7.47	25.87	32.88	69.55
Project Off	3.69	7.47	14.11	10.01	35.28
CM/QA	1.48	2.49	16.46	12.87	33.30
Manuals	1.84	5.81	11.76	10.01	29.42
TOTAL	36.89	83.01	235.20	142.97	498.08

I-MODIFIED TIMELINE SW

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	34.6	6.3	$0	5.5
Product Design	77.8	6.3	$0	12.4
Programming	220.5	6.3	$0	35.1
Integration & Testing	134.0	4.9	$0	27.5
TOTAL	467.0	23.7	$0	19.7

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

SOFTWARE ACTIVITIES	SOFTWARE PHASES				
	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	15.22	7.78	6.62	2.68	32.30
Product Des	5.19	32.69	13.23	5.36	56.47
Programming	2.77	10.12	121.28	58.98	193.15
Test Plng	1.73	5.45	15.44	5.36	27.98
V & V	3.11	7.00	24.26	30.83	65.20
Project Off	3.46	7.00	13.23	9.38	33.08
CM/QA	1.38	2.33	15.44	12.06	31.22
Manuals	1.73	5.45	11.03	9.38	27.59
TOTAL	34.59	77.83	220.52	134.04	466.98

J-MODIFIED ORBIT ANALYSIS SW

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	9.1	3.6	$0	2.5
Product Design	20.5	3.9	$0	5.3
Programming	61.6	4.6	$0	13.5
Integration & Testing	31.9	3.0	$0	10.8
TOTAL	123.1	15.0	$0	8.2

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

SOFTWARE ACTIVITIES	SOFTWARE PHASES				
	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	4.20	2.05	1.85	0.64	8.73
Product Des	1.28	8.62	3.69	1.28	14.87
Programming	0.55	2.46	33.86	12.77	49.64
Test Plng	0.36	1.23	3.69	1.28	6.57
V & V	0.73	1.64	6.16	7.98	16.51
Project Off	1.09	2.26	4.31	2.55	10.21
CM/QA	0.36	0.62	4.31	2.87	8.16
Manuals	0.55	1.64	3.69	2.55	8.44
TOTAL	9.12	20.52	61.56	31.92	123.13

K-MODIFIED DATA FLOW SW

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	46.3	6.9	$0	6.7
Product Design	104.2	6.9	$0	15.1
Programming	295.2	6.9	$0	42.8
Integration & Testing	179.4	5.4	$0	33.5
TOTAL	625.1	26.0	$0	24.0

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

	SOFTWARE PHASES				
SOFTWARE ACTIVITIES	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	20.37	10.42	8.86	3.59	43.24
Product Des	6.95	43.76	17.71	7.18	75.59
Programming	3.70	13.54	162.36	78.95	258.56
Test Plng	2.32	7.29	20.66	7.18	37.45
V & V	4.17	9.38	32.47	41.27	87.29
Project Off	4.63	9.38	17.71	12.56	44.28
CM/QA	1.85	3.13	20.66	16.15	41.79
Manuals	2.32	7.29	14.76	12.56	36.93
TOTAL	46.31	104.19	295.20	179.44	625.13

OUTPUT SECTION 2 - PHASE DISTRIBUTION

SOFTWARE PHASE	EFFORT man-months	SCHEDULE months	COST $	AVERAGE STAFF
Plans & Requirements	3.7	2.7	$0	1.4
Product Design	8.4	2.9	$0	2.9
Programming	25.1	3.4	$0	7.3
Integration & Testing	13.0	2.2	$0	5.9
TOTAL	50.2	11.3	$0	4.5

OUTPUT SECTION 3 - ACTIVITY & PHASE DISTRIBUTION

	SOFTWARE PHASES				
SOFTWARE ACTIVITIES	Plans & Reqts	Product Design	Programming	Integration and Testing	TOTAL
Reqts Anlys	1.71	0.84	0.75	0.26	3.56
Product Des	0.52	3.51	1.51	0.52	6.06
Programming	0.22	1.00	13.80	5.21	20.24
Test Plng	0.15	0.50	1.51	0.52	2.68
V & V	0.30	0.67	2.51	3.25	6.73
Project Off	0.45	0.92	1.76	1.04	4.16
CM/QA	0.15	0.25	1.76	1.17	3.33
Manuals	0.22	0.67	1.51	1.04	3.44
TOTAL	3.72	8.37	25.10	13.01	50.20

END OF OUTPUT SECTION 3